Manuscript Number: IJHCS-D-15-00330R2

Title: Balancing Noise Sensitivity, Response Latency, and Posture Accuracy for a Computer-Assisted Canine Posture Training System

Article Type: SI: ACI

Keywords: Animal-Computer Interaction; Classification algorithm; Machine learning; Random Forest; Wearable technology

Corresponding Author: Mr. John Majikes,

Corresponding Author's Institution: North Carolina State Univeristy

First Author: John Majikes

Order of Authors: John Majikes; Rita Brugarolas; Michael Winters; Sherrie Yuschak; Sean Mealin; Katherine Walker; Pu Yang; Alper Bozkurt; Barbara Sherman; David L Roberts

Manuscript Region of Origin:

Abstract: This paper describes a canine posture detection system comprised of wearable sensors and instrumented devices that detect the postures sit, stand, and eat. The system consists of a customized harness outfitted with wearable Inertial Measurement Units (IMUs) and a base station for processing IMU data to classify canine postures. Research in operant conditioning, the science of behavior change, indicates that successful animal training requires consistent and accurate feedback on behavior. Properly designed computer systems excel at timeliness and accuracy, which are two characteristics most amateur trainers struggle with and professionals strive for. Therefore, in addition to the system being ergonomically designed to ensure the dog's comfort and well-being, it is engineered to provide posture detection with timing and accuracy on par with a professional trainer. We contend that providing a system with these characteristics will one day aid dogs in learning from humans by overcoming poor or ineffective timing during training. We present the initial steps in the development and validation of a computer-assisted training system designed to work outside of laboratory environments.

The main contributions of this work are a) to explore the trade-off between low-latency responses to changes in time-series IMU data representative of posture changes while maintaining accuracy and timing similar to a professional trainer, and b) to provide a model for future ACI technologies by documenting the user-centered approach we followed to create a computer-assisted training system that met the criteria identified in (a). Accordingly, in addition to describing our system, we present the results of three experiments to characterize the performance of the system at capturing sit postures of dogs and providing timely reinforcement. These trade-offs are illustrated through the comparison of two algorithms. The first is Random Forest classification and the second is an algorithm which uses a Variance-based Threshold for classification of postures. Results indicate that with proper parameter tuning, our system can successfully capture and reinforce postures to provide computer-assisted training of dogs.

Balancing Noise Sensitivity, Response Latency, and Posture Accuracy for a Computer-Assisted Canine Posture Training System

John Majikes^{a,1,*}, Rita Brugarolas^{b,1,*}, Michael Winters^a, Sherrie Yuschak^c, Sean Mealin^a, Katherine Walker^b, Pu Yang^a, Barbara Sherman^c, Alper Bozkurt^b, David L. Roberts^a

^a*Department of Computer Science*
^b*Department of Electrical and Computer Engineering*
^c*Department of Clinical Sciences*
North Carolina State University, Raleigh, NC, USA

Abstract

This paper describes a canine posture detection system comprised of wearable sensors and instrumented devices that detect the postures sit, stand, and eat. The system consists of a customized harness outfitted with wearable Inertial Measurement Units (IMUs) and a base station for processing IMU data to classify canine postures. Research in operant conditioning, the science of behavior change, indicates that successful animal training requires consistent and accurate feedback on behavior. Properly designed computer systems excel at timeliness and accuracy, which are two characteristics most amateur trainers struggle with and professionals strive for. Therefore, in addition to the system being ergonomically designed to ensure the dog’s comfort and well-being, it is engineered to provide posture detection with timing and accuracy on par with a professional trainer. We contend that providing a system with these characteristics will one day aid dogs in learning from humans by overcoming poor or ineffective timing during training. We present the initial steps in the development and validation of a computer-assisted training system designed to work outside of laboratory environments.

The main contributions of this work are a) to explore the trade-off between low-latency responses to changes in time-series IMU data representative of posture changes while maintaining accuracy and timing similar to a professional trainer, and b) to provide a model for future ACI technologies by documenting the user-centered approach we followed to create a computer-assisted training system that met the criteria identified in (a). Accordingly, in addition to describing our system, we present the results of three experiments to characterize the performance of the system at capturing sit postures of dogs and providing timely reinforcement. These trade-offs are illustrated through the comparison of two algorithms. The first is Random Forest classification

*Corresponding authors:
Email addresses: jjmajike@ncsu.edu (John Majikes), rbrugar@ncsu.edu (Rita Brugarolas)
¹These authors contributed equally to this work.

and the second is an algorithm which uses a Variance-based Threshold for classification of postures. Results indicate that with proper parameter tuning, our system can successfully capture and reinforce postures to provide computer-assisted training of dogs.

Keywords: Animal-Computer Interaction; Classification algorithm; Machine learning; Random Forest; Wearable technology

1. Introduction

As long as dogs have been domesticated, humans have spent time and effort to train them to perform a multitude of tasks. Research in applied behavior analysis suggests the most effective approach to animal training emphasizes the use of positive reinforcement [12], which involves giving something the animal desires such as food to reward the sought-after behavior. Training is a process by which animals learn to perform specific, desired behaviors when instructed. One common training procedure is to allow an animal to produce a desired natural behavior, such as sit. Then, the animal is immediately rewarded with a primary reinforcer (a food treat), delivered by the human trainer. Over time, the rewarded behavior (sit) occurs with increasing frequency. Then, the auditory cue, “*sir*”, is given by the trainer antecedent to the behavior and an association between the cue and action is established. Training is often slowed due to inadequate knowledge or skill on the part of the trainer. Fundamental concepts that affect the animal’s performance are *timing*, *criteria*, and *rate of reinforcement* [1]. Human trainers, especially novices, often struggle to provide timely and consistent reinforcement when teaching animals. Indiscriminately reinforcing multiple behaviors or failing to recognize correct behaviors are examples of inadequate training criteria and can slow down the learning process, while reinforcing the wrong behavior can teach the dog to do an undesirable action. Timing is critical: Reinforcement is most effective when delivered as close as possible to the desired behavior; a delay greater than 0.5 seconds can retard a dog’s ability to create the association between the instruction, the behavior, and the reinforcement [25].

Computers, on the other hand, excel at extremely accurate timing and maintaining consistent criteria and could be used to enhance the process of training animals and by reducing human error, might reduce ambiguity and improve the human-canine relationship. Thus, human-animal-computer interactions could enhance our inter-species relationships with the animals we live or work with [14]—a computer training system would improve communication and benefit the dog. Since inadequate or ineffective training may lead to dogs being relinquished to animal shelters, a semi-autonomous computer training system for dogs could help eliminate training issues which would help reduce the number of dogs sent to shelters. In addition, a semi-autonomous training system could be used in animal shelters to train dogs to provide beneficial stimulation[15] and to enhance adoptability. Research has shown that shelter dogs that sit or stand in response to interest from a potential adopter are more likely to be adopted [22]. Our future research work would be to test this system on shelter dogs where human involvement in training might be costly or logistically prohibitive.

This paper introduces a canine posture detection system comprised of wearable sensors and instrumented devices that detect canine postures such as sit, stand, or eat. The system consists of a customized harness outfitted with wearable Inertial Measurement Units (IMUs) and a base station for processing IMU data to classify canine posture.

The latency and accuracy of feedback are extremely important when training animals. Research in operant conditioning, the science of behavior change, indicates that successful animal training requires consistent and accurate feedback on behavior. Properly designed computer systems excel at timeliness and accuracy, which are two characteristics most amateur trainers struggle with and professionals strive for. Therefore, we present initial steps in the development and validation of a computer-assisted training system designed to work outside of laboratory environments.

The main contributions of this work are to explore the trade-off between low-latency responses to changes in time-series IMU data representative of posture changes while maintaining accuracy and timing similar to a professional trainer. Accordingly, in addition to describing our system, we present the results of three experiments to characterize the performance of the system at capturing sit postures of dogs and providing timely reinforcement. These trade-offs are illustrated through the comparison of two algorithms. The first is Random Forest classification and the second is an algorithm which uses a Variance-based Threshold for classification of postures. Results indicate that with proper parameter tuning, our system can successfully capture and reinforce postures to provide computer-assisted training of dogs.

International Journal of Human - Computer Studies

AUTHOR AGREEMENT FORM

Manuscript Title: Balancing Noise Sensitivity, Response Latency,
and Posture
Accuracy for a Computer-Assisted Canine Posture Training
System

List of All Authors: John Majikes^{a,1,_,} Rita Brugarolas^{b,1,_,} Michael Winters^{a,} Sherrie
Yuschak^{c,} Sean
Mealina^{a,} Katherine Walker^{b,} Pu Yang^{a,} Alper Bozkurt^{b,} Barbara Sherman^{c,} David L.
Roberts^a

Corresponding Author: John J. Majikes, Rita Brugarolas

This statement is to certify that that the author list is correct, all Authors have seen and approved the manuscript being submitted and agree to its submission to the *International Journal of Human - Computer Studies*. The Authors also confirm that this research has not been published previously and that it is not under consideration for publication elsewhere. On behalf of all Co-Authors, the Corresponding Author shall bear full responsibility for the submission.

All authors agree that the author list is correct in its content and order and that no modification to the author list can be made without the formal approval of the Editor-in-Chief. All authors accept that the Editor-in-Chief's decisions over acceptance, rejection or retraction (the latter in the event of any breach of the Principles of Ethical Publishing in the International Journal of Human - Computer Studies being discovered) are final.

07/17/15

To whom it may concern:

On behalf of my coauthors and myself, I am pleased to offer the submission titled “Balancing Noise Sensitivity, Response Latency, and Posture Accuracy for a Computer-Assisted Canine Posture Training System.” This paper contains substantive original work in the area of canine posture detection, and computer-assisted training of canines. The work contained in this paper is not currently under review for any other publication venue.

Sincerely,

John J. Majikes
1904 Steam Ridge Ct.
Apex, NC 27523 USA

Figure

[Click here to download high resolution image](#)

Figure

[Click here to download high resolution image](#)

As long as dogs have been domesticated, humans have spent time and effort to train them to perform a multitude of tasks. Research in applied behavior analysis suggests the most effective approach to animal training emphasizes the use of positive reinforcement [12], which involves giving something the animal desires such as food to reward the sought-after behavior. Training is a process by which animals learn to perform specific, desired behaviors when instructed. One common training procedure is to allow an animal to produce a desired natural behavior, such as sit. Then, the animal is immediately rewarded with a primary reinforcer (a food treat), delivered by the human trainer. Over time, the rewarded behavior (sit) occurs with increasing frequency. Then, the auditory cue, “sit”, is given by the trainer antecedent to the behavior and an association between the cue and action is established. Training is often slowed due to inadequate knowledge or skill on the part of the trainer. Fundamental concepts that affect the animal’s performance are timing, criteria, and rate of reinforcement [1]. Human trainers, especially novices, often struggle to provide timely and consistent reinforcement when teaching animals. Indiscriminately reinforcing multiple behaviors or failing to recognize correct behaviors are examples of inadequate training criteria and can slow down the learning process, while reinforcing the wrong behavior can teach the dog to do an undesirable action. Timing is critical: Reinforcement is most effective when delivered as close as possible to the desired behavior; a delay greater than 0.5 seconds can retard a dog’s ability to create the association between the instruction, the behavior, and the reinforcement [25].

Computers, on the other hand, excel at extremely accurate timing and maintaining consistent criteria and could be used to enhance the process of training animals and by reducing human error, might reduce ambiguity and improve the human-canine relationship. Thus, human-animal-computer interactions could enhance our inter-species relationships with the animals we live or work with [14]—a computer training system would improve communication and benefit the dog. Since inadequate or ineffective training may lead to dogs being relinquished to animal shelters, a semi-autonomous computer training system for dogs could help eliminate training issues which would help reduce the number of dogs sent to shelters. In addition, a semi-autonomous training system could be used in animal shelters to train dogs to provide beneficial stimulation[15] and to enhance adoptability. Research has shown that shelter dogs that sit or stand in response to interest from a potential adopter are more likely to be adopted [22]. Our future research work would be to test this system on shelter dogs where human involvement in training might be costly or logistically prohibitive.

Accordingly, we have developed a semi-autonomous training system consisting of a smart harness outfitted with inertial measurement units (IMUs) worn by a dog, a laptop computer that collects data from the IMUs, algorithms for classifying postures, and a computer-controlled treat dispenser to provide reinforcement in response to desired behaviors. An illustration of the system appears in Figure 1. In this paper, as a first step towards achieving real-world computer-assisted training, we address the issues of creating a system to provide accurate timing of reinforcement and maintaining consistent behavioral criteria. In computational terms, these two factors directly translate to response latency and noise sensitivity when designing algorithms for capturing and reinforcing postures.

In the remainder of this paper we will discuss how knowledge of behavior analysis has informed the performance and ergonomic design criteria for our system, some of the practical issues in the design of a computer-assisted canine training system, the efforts made to ensure the dog’s comfort and acceptance of the system, and present the hardware and software we have developed in some detail. We will then present the results of three experiments that address:

1. algorithmic noise sensitivity and response latency in the context of data preprocessing,
2. timing and accuracy in comparison to a professional trainer, and
3. evidence of dogs learning as it relates to timing and accuracy.

The first experiment illustrates how response latency and noise sensitivity must be managed effectively through data pre-processing to achieve a balance between timely and accurate feedback. The second experiment includes a more extensive validation of the timing experiments using a professional dog trainer as a point of comparison. Finally, the third experiment is intended to further validate the effectiveness of the system by showing the conditions under which it successfully (and unsuccessfully) is able to semi-autonomously teach a dog to sit for food rewards.

Balancing Noise Sensitivity, Response Latency, and Posture Accuracy for a Computer-Assisted Canine Posture Training System

John Majikes^{a,1,*}, Rita Brugarolas^{b,1,*}, Michael Winters^a, Sherrie Yuschak^c, Sean Mealin^a, Katherine Walker^b, Pu Yang^a, Barbara Sherman^c, Alper Bozkurt^b, David L. Roberts^a

^a*Department of Computer Science*

^b*Department of Electrical and Computer Engineering*

^c*Department of Clinical Sciences*

North Carolina State University, Raleigh, NC, USA

Abstract

This paper describes a canine posture detection system comprised of wearable sensors and instrumented devices that detect the postures sit, stand, and eat. The system consists of a customized harness outfitted with wearable Inertial Measurement Units (IMUs) and a base station for processing IMU data to classify canine postures. Research in operant conditioning, the science of behavior change, indicates that successful animal training requires consistent and accurate feedback on behavior. Properly designed computer systems excel at timeliness and accuracy, which are two characteristics most amateur trainers struggle with and professionals strive for. Therefore, in addition to the system being ergonomically designed to ensure the dog's comfort and well-being, it is engineered to provide posture detection with timing and accuracy on par with a professional trainer. We contend that providing a system with these characteristics will one day aid dogs in learning from humans by overcoming poor or ineffective timing during training. We present the initial steps in the development and validation of a computer-assisted training system designed to work outside of laboratory environments.

The main contributions of this work are a) to explore the trade-off between low-latency responses to changes in time-series IMU data representative of posture changes while maintaining accuracy and timing similar to a professional trainer, and b) to provide a model for future ACI technologies by documenting the user-centered approach we followed to create a computer-assisted training system that met the criteria identified in (a). Accordingly, in addition to describing our system, we present the results of three experiments to characterize the performance of the system at capturing sit postures of dogs and providing timely reinforcement. These trade-offs are illustrated through the comparison of two algorithms. The first is Random Forest classification

*Corresponding authors:

Email addresses: jjmajike@ncsu.edu (John Majikes), rbrugar@ncsu.edu (Rita Brugarolas)

¹These authors contributed equally to this work.

and the second is an algorithm which uses a Variance-based Threshold for classification of postures. Results indicate that with proper parameter tuning, our system can successfully capture and reinforce postures to provide computer-assisted training of dogs.

Keywords: Animal-Computer Interaction; Classification algorithm; Machine learning; Random Forest; Wearable technology

1. Introduction

As long as dogs have been domesticated, humans have spent time and effort to train them to perform a multitude of tasks. Research in applied behavior analysis suggests the most effective approach to animal training emphasizes the use of positive reinforcement [?], which involves giving something the animal desires such as food to reward the sought-after behavior. Training is a process by which animals learn to perform specific, desired behaviors when instructed. One common training procedure is to allow an animal to produce a desired natural behavior, such as sit. Then, the animal is immediately rewarded with a primary reinforcer (a food treat), delivered by the human trainer. Over time, the rewarded behavior (sit) occurs with increasing frequency. Then, the auditory cue, “sit”, is given by the trainer antecedent to the behavior and an association between the cue and action is established. Training is often slowed due to inadequate knowledge or skill on the part of the trainer. Fundamental concepts that affect the animal’s performance are timing, criteria, and rate of reinforcement [?]. Human trainers, especially novices, often struggle to provide timely and consistent reinforcement when teaching animals. Indiscriminately reinforcing multiple behaviors or failing to recognize correct behaviors are examples of inadequate training criteria and can slow down the learning process, while reinforcing the wrong behavior can teach the dog to do an undesirable action. Timing is critical: Reinforcement is most effective when delivered as close as possible to the desired behavior; a delay greater than 0.5 seconds can retard a dog’s ability to create the association between the instruction, the behavior, and the reinforcement [?].

Computers, on the other hand, excel at extremely accurate timing and maintaining consistent criteria and could be used to enhance the process of training animals and by reducing human error, might reduce ambiguity and improve the human-canine relationship. Thus, human-animal-computer interactions could enhance our inter-species relationships with the animals we live or work with [?]—a computer training system would improve communication and benefit the dog. Since inadequate or ineffective training may lead to dogs being relinquished to animal shelters, a semi-autonomous computer training system for dogs could help eliminate training issues which would help reduce the number of dogs sent to shelters. In addition, a semi-autonomous training system could be used in animal shelters to train dogs to provide beneficial stimulation [?] and to enhance adoptability. Research has shown that shelter dogs that sit or stand in response to interest from a potential adopter are more likely to be adopted [?]. Our future research work would be to test this system on shelter dogs where human involvement in training might be costly or logistically prohibitive.

Figure 1: The Computer-Assisted Training platform, including a smart harness with IMUs, a laptop with algorithms for posture detection, and a computer-controlled treat dispenser to reinforce desirable behavior.

Accordingly, we have developed a semi-autonomous training system consisting of a smart harness outfitted with inertial measurement units (IMUs) worn by a dog, a laptop computer that collects data from the IMUs, algorithms for classifying postures, and a computer-controlled treat dispenser to provide reinforcement in response to desired behaviors. An illustration of the system appears in Figure 1. In this paper, as a first step towards achieving real-world computer-assisted training, we address the issues of creating a system to provide accurate timing of reinforcement and maintaining consistent behavioral criteria. In computational terms, these two factors directly translate to *response latency* and *noise sensitivity* when designing algorithms for capturing and reinforcing postures.

In the remainder of this paper we will discuss how knowledge of behavior analysis has informed the performance and ergonomic design criteria for our system, some of the practical issues in the design of a computer-assisted canine training system, the efforts made to ensure the dog’s comfort and acceptance of the system, and present the hardware and software we have developed in some detail. We will then present the results of three experiments that address:

1. algorithmic noise sensitivity and response latency in the context of data pre-processing,
2. timing and accuracy in comparison to a professional trainer, and
3. evidence of dogs learning as it relates to timing and accuracy.

The first experiment illustrates how response latency and noise sensitivity must be managed effectively through data pre-processing to achieve a balance between timely and accurate feedback. The second experiment includes a more extensive validation of the timing experiments using a professional dog trainer as a point of comparison. Finally, the third experiment is intended to further validate the effectiveness of the system by showing the conditions under which it successfully (and unsuccessfully) is able to semi-autonomously teach a dog to sit for food rewards.

2. Related Work

The use of wearable inertial measurement units on humans has become increasingly popular. Studies of these units use a wide variety of machine learning algorithms to classify postures [?]. There has been work in applying similar classification techniques using sensors to animals [? ? ?]; however, these studies have primarily focused on monitoring activity levels of animals, not identifying specific activities [?]. While using accelerometers to measure postures is not entirely new, there are two aspects of our approach that are novel. First, our system is designed for computer-assisted training, and therefore closes the loop from posture recognition to feedback. Second, our system meets a multi-criteria performance objective: that accuracy is equally important as response latency.

Posture recognition uses inertial sensors to classify the activity the dog is currently performing. In one study, recognition was done in real-time; however the positioning of the sensors were placed inefficiently for posture recognition [?]. In prior work, we identified positions for sensors on a canine in order to optimize for posture classification accuracy (without attention to latency) [?]. In other work, accelerometer data was used to identify when and for how long canines exhibited a total of seven static postures and dynamic behaviors, however the recognition was not done in real-time due to the reliance on data that needed to be manually extracted from video, and the need to connect to the sensors to get the data [?]. Our previous work on the evaluation of a machine learning algorithm based on a two-stage cascade classifier used raw sensor data [?] to accurately recognize five static postures and three dynamic behaviors in near real-time [?]. To our knowledge there is no existing work examining the trade-offs between noise, response latency, and posture accuracy of real-time classification algorithms as compared to a professional trainer (as reported in this paper).

2.1. A Platform for Computer-Assisted Training

To realize computer-assisted training, we’ve developed both hardware and software systems that provide a real-time view of canine postures and the capability to reinforce those postures with minimal human input. The system consists of a custom harness outfitted with Inertial Measurement Units (IMUs) on the back and chest area of the dog, a base unit running classification algorithms, and a remotely operated treat dispenser. To close the loop and provide rewards to the dog for the desirable postures, remotely-operated treat dispensers are triggered based on the output of the classification algorithms. At this point in time the system is not a fully-autonomous since human input is needed to algorithmically train the posture recognition algorithms. Reducing human involvement is a subject of ongoing work [?]. An overview picture of our system is included in Figure 1.

Drawing on knowledge of applied behavior analysis and canine physiology, we identified the following *design criteria* for our system:

1. **Ergonomics:** In accordance with Mancini’s Animal-Computer Interaction (ACI) Manifesto [?], the dogs comfort and psychological well-being were the most important design criteria. Any system we developed first and foremost was designed to “*Protect both human and nonhuman participants from physiological*

or psychological harm at all times...” [?]. Accordingly, our design process included a careful evaluation of the ergonomics of the hardware platform, including weight, position, and size. We iteratively refined the design by working with a veterinary technician specialized in animal behavior to identify any signs of stress, discomfort, or behavioral inhibition. Our process is detailed in Section 2.3.

2. **Timing:** Research in Applied Behavior Analysis, the study of how behavior changes in animals, has demonstrated the significance of the delay between a behavior (posture change in our case) and the delivery of reinforcement (conditioned or primary) [?]. As a general rule, it is known that the shorter the latency between the desired behavior and the delivery of the reinforcement to the animal, the faster and more accurate the learning. In particular, a delay of more than 0.5 seconds significantly hinders learning in dogs [?]. Therefore, our goal is a 0.2 second delay in our system. Our efforts are detailed in Sections 3.2.2 & 3.3.2.
3. **Consistency:** The Matching Law of Behavior [?] is a guiding principle of animal training that dictates behaviors are influenced by how often they are reinforced. It follows that it is critically important for our system to be as accurate as possible, so as to maintain consistent criteria for what is reinforced. Operationally, this translated to a goal of having as few erroneous (either additions or deletions) posture recognitions as possible.

In general, all three of these criteria are related. Ergonomic considerations dictate where sensors can be placed on dogs, which may affect the ability of algorithms to detect postures. Similarly, the speed of detection can be influenced by the tolerance of the algorithms for noise in the data. Throughout the remainder of this paper, we will detail the hardware, ergonomic, and software designs we created in order to realize a computer-assisted training system that meets the needs of dogs as defined by these criteria.

2.2. Hardware

The main part of our system is the smart harness, the development of which has been detailed in our prior work [?]. Based on those results, we made several improvements to the previous smart harness that are described Section 2.3. For continuity, we provide a brief description of the previous harness before describing the improvements.

To handle sensor information and communications, the harness is equipped with a small BeagleBone Black (BBB) computer. The BBB includes a 1-GHz processor from Texas Instruments (TI; AM3358BZCZ100), 2 Gbytes of on-board flash storage, and 512 Mbytes of DDR3 RAM. It also includes up to 65 general-purpose input/output accessible pins, eight pulse-width modulation (PWM) channels, and eight channels of 12-bit analog to digital converters and 2 digit I2C serial buses, which are used to interface with the two IMUs. The two IMUs each included a three-axis accelerometer (LSM303) and three-axis gyroscope (L3GD20H) and were configured to produce readings at 10Hz. The BBB runs Ubuntu GNU/Linux, giving access to the utilities and development tools found on a standard GNU/Linux system. The communication link is IEEE 802.11, and most of the communication to and from the base station is done using User Datagram Protocol (UDP) to increase speed. We included a watchdog

program that is responsible for monitoring the wireless connection and other services, and restarting them if they encounter an error or fail to respond.

The base unit is a Lenovo Thinkpad W530 with an Intel Core i7-374QM 2.7 GHz processor, 8GB RAM, running 64-bit Windows 7. The base unit processed the sensor data, took human input for labeling sensor data, ran algorithms for determining posture (described below), and provided data logging capabilities. We integrated a laser pointer (with the laser disabled) to function as a hand-held remote-control to communicate a professional trainer's interpretation of posture changes. The laser pointer allowed our professional trainer to operate much like they would using a clicker (a behavior marker and also a conditioned reinforcer [?]). Using the remote, we can log their timing data for comparison to our algorithms.

2.3. Harness Ergonomics

In prior work we modified a Lift Load Carry Harness from Ray Allen Manufacturing (SKU RA36MHL-P) to include hardware for sensing and communications. The harness has three straps, one around the neck, one around the chest, and one around the belly. It also contains Velcro™ for easy attachment of accessories. Before beginning the experiments the prior work's smart harness was modified to be more compatible and comfortable for the range of dogs used in the testing. Considering that many dogs of varying age, weight, and physical ability should be able to benefit from our efforts, we modified the harness to be more comfortable and less bulky. Under the observation of a veterinary technician specialized in animal behavior who is credentialed as a Certified Professional Dog Trainer-Knowledge Assessed (CPDT-KA), a Veterinary Behavior Technician (VBT), a Registered Veterinary Technician (RVT), a Veterinary Technician Specialist (VTS) in behavior, and has more than 15 years of dog training experience, the fitting of the harness for each dog included treats during a slow introduction period. All dogs became comfortable within several minutes using this introduction method.

One important consideration was the weight of the harness. During discussions with the VBT we learned that untrained dogs can safely carry between 5% and 10% of their body weight. The prior work's harness contained two large batteries (0.3 kg each), one on each side of the harness. The original intent was to have the batteries weight down both sides of the harness to maintain a more stable position for the IMUs. This harness weighed approximately 1.6 kg and would have been too heavy for smaller dogs. Algorithm 1 was found to tolerate small movement of the harness such that a single 0.1 kg battery could be centered on the harness and still provide sufficient stability for the experiments. The finalized harness weighed 0.7 kg.

In addition to the weight and size, the prior work's harness was secured around the neck by a double-pass steel buckle. Adjusting the buckle was slow and the process made some dogs uncomfortable, so we fixed the buckle in place. To wear the harness the neck strap was simply placed over the dog's head. Preliminary testing of the harness on our own dogs showed that some dogs backed away when the harness was above the dog's head. To accommodate this perceived discomfort, the original harness double-pass buckle was modified to use a Velcro™ fastener around the neck that opened quickly and secured more easily than the original. Using the Velcro™ fastening neck strap, the dogs showed no backward stepping in any of the experiments.

2.4. Harness Accelerometer Position

When developing wearable technologies for canines there are a limited number of **practical** sensor sites. For example most dogs will not tolerate sensors strapped to their paws, tail, or around their snout. Because comfort for the dogs was a primary concern for this work, we limited our investigation to areas of the body our **VBT** indicated would be least likely to cause discomfort.

Our previous research [? ?] showed how we optimized inertial sensor sites by considering the kinetics of the canine to identify independently moving locations on the body. Four locations were tested: the chest of the animal, the abdomen, and two locations at the back; one close to the head (around withers) and the other close of the tail (around rump) of the animal.

Since accelerometers measure not only the dynamic acceleration due to actual motion, but also the static acceleration, which corresponds to the projection of gravity over the axes of the sensor, different postures may result in similar data depending on how the IMUs are aligned with respect to the dog’s body and gravity, thereby requiring a multi-sensor measurement to assess the posture of the dog accurately.

In order to select the locations we looked at the angle of change along three axes of accelerometer data for different postures relative to the baseline standing posture. For example as it was suggested by others [?], the two sensors on the back provide very similar sensor data due to both sites moving together during most postures.

The optimal locations for IMUs leading to larger angle changes between postures were observed when sensors were placed at the chest of the animal and at the end of the back around the rump. These location fit our ergonomic design goals.

2.5. Classification Algorithms

The job of the classification algorithms is to turn the IMU readings coming from the harness at 10Hz into labels of postures as quickly and accurately as possible. After experimentation on posture detection in earlier work, we determined that a Random Forrest (RF) classifier produced extremely accurate results [? ? ?]; however, that work focused solely on accuracy, and did not account for latency. Accordingly, we chose to use the RF classifier in this work because it has been highly-accurate in the past and, due to its simplicity (relative to other machine learning algorithms), it is efficient.

To better understand the relationship between accuracy and latency, we employed two techniques: 1) data smoothing, and 2) a very simple threshold-based classification scheme. Note that we make no claim about these techniques being optimal for training—that would require much more extensive empirical testing; however, because of their varying performance characteristics which we’ll discuss below, they provide a good counterpoint for examining the influence of accuracy and latency in a computer-assisted training system. It remains an open question as to whether there is an “optimal” classification scheme.

To understand these concepts, consider Figure 2. The Figure contains a plot from our earlier work [?] where five different postures were considered: standing, laying down, sitting, standing on two legs, and eating off the floor. The x-axis in the plot contains the IMU reading from the dog’s chest parallel to gravity, and the y-axis of the plot contains the IMU reading from the dog’s back near the base of its tail perpendicular

Figure 2: A plot of x-axis acceleration from two different IMUs for five different postures.

to gravity, aligned with its spine. In all five postures, the y-axis of the figure is relatively tightly grouped; however, especially for standing and eating off the floor, the x-axis of the figure has higher variance due to the vertical movement of the head while in these postures. This example illustrates one type of noise in the data, which may be from multiple sources. One approach to addressing noise is to filter the data using a moving average filter [?]. The more aggressively the data are filtered, the more noise gets removed; however, as a side effect, true changes in IMU readings are slower to be detected—a higher latency. We will present data to illustrate the effects of this trade-off below.

The second approach we investigated for addressing the latency vs. accuracy trade-off was to implement a very efficient threshold classification scheme. “Variance-based threshold” classification (VT), works by identifying the means and standard deviations of each of the six IMU axes, identifying boundaries between those values that differentiated the data associated with different postures, and constructing a set of decision rules accordingly. For example, examining the data in Figure 2, a value of the x-acceleration on the front IMU (x-axis in the Figure) below 0.4 is very-highly-correlated with the eating off the floor posture. One way to interpret the VT technique is as computing probabilistic bounding boxes around the IMU readings associated with different postures. For all measurements, we use VT classification to imply the classification of values within a range of the mean plus or minus a multiple of the standard deviation.

If the sensor values for the five postures are normally distributed we could predict the likelihood of future values falling within certain ranges. For example, if the data ranges between $\mu \pm 3.0\sigma$ for one posture (say eating) and no other posture results in data in that range, then we could be 99.7% confident that a sensor reading in that range reflects a eating posture. Given that the sensor readings from multiple sources are not independent and normally-distributed sources of data we cannot use these estimates formally; however, it does give us a general preference for selecting as large a σ range

Algorithm 1 VT Threshold Calculation

```

1:  $P = \{ \text{sit, stand, eat, ...} \}$  # Set of postures
2:  $S = \{x_1, y_1, z_1, x_2, y_2, z_2\}$  # Set of sensors
3: Let  $L = \{x_1, y_1, z_1, x_2, y_2, z_2, p\}$  # Labeled data is sensor data and observed posture ( $p \in P$ )
4: Let  $L_{s,p} | p \in P, s \in S$  # Sensor values for a specific sensor and posture
5:
6: # Calculate the mean and standard deviation for each set of sensor and posture pairings
7:  $\mu_{s,p} = \text{mean } L_{s,p}$ 
8:  $\sigma_{s,p} = \text{stdev } L_{s,p}$ 
9:
10: # Calculate  $\mu\text{Diff}$ , the difference between the means of two postures for a given sensor
11: # Calculate  $\sigma\text{Distance}$ , the  $\mu\text{Diff}$  expressed in units of standard deviation
12: # A  $\sigma\text{Distance}_{s,p_1,p_2} = 3$  implies a strict separating range out to three standard deviations.
13: for all  $p' \in P$  do
14:   for all  $p'' \in P, p' \neq p''$  do
15:     for all  $s \in \{x_1, y_1, z_1, x_2, y_2, z_2\}$  do
16:        $\mu\text{Diff}_{s,p',p''} = |\mu_{s,p'} - \mu_{s,p''}|$ 
17:        $\sigma\text{Distance}_{s,p',p''} = \frac{\mu\text{Diff}_{s,p',p''}}{\sigma_{s,p'} + \sigma_{s,p''}}$ 
18:     end for
19:   end for
20: end for
21:
22: # Find the minimum  $\sigma\text{Distance}$  for each sensor and posture.
23: Let  $\mu\text{Min}_{s,p} = \min(\sigma\text{Distance}_{s,p,p'}) | p' \in P, p \neq p'$ 
24:
25: # For each posture, find the two largest  $\mu\text{Min}$  that will be used for classification.
26: for all  $p \in P$  do
27:    $\text{largestMuMin}_p, \text{secondLargestMuMin}_p = 0$ 
28:    $\text{largestMuMinSensor}_p, \text{secondLargestMuMinSensor}_p = 0$ 
29:   for all  $s' \in \{x_1, y_1, z_1, x_2, y_2, z_2\}$  do
30:     if  $\mu\text{Min}_{s',p} > \text{largestMuMin}_p$  then
31:       # Remember the new largest and second largest ranges
32:        $\text{secondLargestMuMin}_p = \text{largestMuMin}_p$ 
33:        $\text{largestMuMin}_p = \mu\text{Min}_{s',p}$ 
34:
35:       # Remember which sensors provide the ranges
36:        $\text{secondLargestMuMinSensor}_p = \text{largestMuMinSensor}_p$ 
37:        $\text{largestMuMinSensor}_p = s'$ 
38:     else
39:       if  $\mu\text{Min}_{s',p} > \text{secondLargestMuMin}_p$  then
40:         # Remember range and sensor that's second largest
41:          $\text{secondLargestMuMin}_p = \mu\text{Min}_{s',p}$ 
42:          $\text{secondLargestMuMinSensor}_p = s'$ 
43:       end if
44:     end if
45:   end for
46: end for
47:
48: return  $\{ \text{largestMuMinSensor}, \text{secondLargestMuMinSensor}, \mu_{s,p}, \sigma_{s,p}, \mu\text{Min} \}$ 

```

as possible, while maintaining strict separation between the ranges associated with each axis for each posture. Algorithm 1 presents the process of finding the largest range of values expressed as a multiple of σ from the μ . These values are used to construct the decision rules in the VT classifier, which are then used to classify postures in real-time.

3. Experiments

We conducted three separate experiments to understand and validate the performance of our system. In all experiments our **VBT** monitored the dogs for any signs of stress, including (but not limited to) panting, lip licking, “whale eye,” pacing, **avoidance of the testing area**, or a lack of interest in the task. The first experiment, presented in Section 3.1, involved collecting posture data from two dogs to identify the optimal parameter settings for pre-processing the data. The second experiment, presented in Section 3.2, involved using the classification algorithms with the parameters identified in the first experiment to compare against the timing and accuracy of a professional dog trainer. Lastly, in Section 3.3, we present the results of a further validation of the system and assumptions about response latency whereby various parameter settings were compared in the context of a semi-autonomous computer training system and evaluated based on evidence of dogs learning.

All **experimental** procedures were approved by the Institutional Animal Care and Use Committee of North Carolina State University (NC State).

3.1. Data Pre-Processing

To address our second and third design criteria of timing and consistency, we experimented with different ways of pre-processing our data. We used a simple moving average filter with varying window sizes to balance noise reduction while limiting increases in latency. None of the dogs used in this experiment were used in the timing experiment in Section 3.2, but the results from this experiment set the moving average window size for the timing experiment.

The moving average window size is a knob that enables us to make a trade-off between noise sensitivity and latency. A small window size allows true changes in sensor values to be reflected more quickly in the data, but the influence of high-frequency noise in the data remains high. A classification algorithm applied to data with a small moving average window size will likely have low latency in detecting postures, but suffer a lower accuracy resulting from the presence of more noise in the data. On the other hand, applying a large moving average window size will result in very clean data, free from noise. This reduction in noise will come at the cost of response latency. By aggressively averaging the data, true changes in sensor readings will also be smoothed, thereby increasing the latency before the posture change gets reflected in the data. The experiments in this section are designed to empirically identify parameter settings that result in the best trade-off between response latency and accuracy over the RF and VT classifiers.

Table 1: Number of sit, stand, and eat data instances along with the duration for dogs used in tuning the moving average window size.

Dog		Sits	Eats	Stands	Total
Dog_α	Instances (10Hz)	920	57	1881	2858
	Time (mm:ss)	01:32.0	00:05.7	03:08.1	04:45.8
Dog_β	Instances (10Hz)	1093	0	510	1603
	Time (mm:ss)	01:49.3	00:00.0	00:51.0	02:40.3

3.1.1. Setup

The experimental process involved collecting data from two dogs, Dog_α and Dog_β . See Table A.7 in the Appendix for more information on the dogs used, including age, breed, sex, and weight. The dogs were fitted with the smart harness, and were led through several repetitions of the three postures. The dogs were given food rewards after maintaining each posture for somewhere between two and ten seconds. We collected between three to five minutes of data from each dog. Table 1 shows for each dog, the number of sit, stand, and eat instances collected.

The x-, y-, and z-axis components of the two accelerometers were collected at 10Hz, and labeled in real-time by a pair of experimenters. For Table 1 Dog_α sat for a total of 920 instances or a combined 92.0 seconds of the 285.8 second experimental period. Both of the experimenters were responsible for watching the dog. One of the experimenters was responsible for calling out the posture changes and the other for entering the label on the keyboard of the base unit. Using two experimenters reduced the chances of error and distraction causing latency in applying the labels.

The RF classifier used in this experiment is from the sklearn RandomForestClassifier.²

To characterize performance of different moving average window sizes, the raw data was post-processed using different moving average window sizes ranging from 1 (no filtering) to 10 (a 1-second average). We performed 10-fold cross validation on each moving average size to compare instance-based accuracy. Since the cross validation was done using instances of data that were Independent and Identically Distributed (IID), instance-based accuracy used in the cross validation is a true test of instance-classification accuracy.

Ultimately the goal is not to detect sit instances as used in 10-fold cross validation but to capture a sit posture so the distinction between stand instances and eat instances is superfluous; however, to ensure our data more closely represented the distribution of postures we will encounter in later experiments, eating was included in the data set as well—taking a treat from the computer-controlled treat dispenser is the same posture as eating off the floor. Results, however, are reported only for two labels: “sitting” and “not sitting” (which includes standing and eating).

Figure 3: Number of FN over varying moving average window sizes. Arrows show where the lowest FN occurred with the smallest window size.

Figure 4: Percent accuracy instance-based classification over varying moving average window sizes. Arrows show where the highest percent instance-classification occurred with the smallest moving average window size.

3.1.2. Results

Figure 3 shows the false negative (FN) for both the VT and RF instance-based classification for Dog_α and Dog_β using moving average sizes of 1 through 10. Figure 4 shows the instance-based accuracy for both classification algorithms for the same dogs and window sizes. The arrows in the figure indicate the smallest moving average window size where the best algorithm performance occurred.

The arrows in Figure 3 show that the smallest window size that had the fewest FN for Dog_α VT and RF instance-based classification, and Dog_β VT and RF instance-based classification are at six, three, three, and two respectively. Similarly, the arrows in Figure 4 show the smallest window size that had the highest percent instance-based classification accuracy for Dog_α VT and RF, and Dog_β VT and RF are at four, three, three, and three respectively.

For a window size of four the accuracy was always as good or better than a window size of three. In addition, except for Dog_α VT, a window size of 4 had the lowest number of FN. Therefore, based solely on accuracy, a window of size four would seem to be the best choice for future experiments; however, to be able to know for sure, latency also has to be considered.

3.2. Timing and Accuracy

The second experiment had two goals. First, the parameters identified in the first experiment were used to compare the timing and accuracy of the classification algorithms. Second, outlier analysis was done to give some indication of the usefulness of the harness with dogs of varying age, weight, and physical ability. To validate the algorithms as being capable of detecting posture effectively, we chose to compare latency and accuracy of our semi-autonomous system with our VBT. To check for the usefulness of the harness with different dogs, Dixons' [?] and Grubbs' [?] tests were used for detecting outliers of the dogs' percentage accuracy.

Since entering a sit posture is a process that requires several anatomical movements by the dog, and to provide a consistent point of comparison for labeling, the three experimenters noted when the dog's rump hit the floor by pressing a button on a keyboard. The average time of the three experimenters' inputs was used as the point of comparison of when the sit occurred. Timing accuracy reported here compares this point in time with the time of the VBT and with the time of the classification algorithms.

Recall that our overarching goal is to achieve computer-assisted training. Training a dog to sit requires capturing the posture by precisely reinforcing at the time at which the sit occurs—this is the posture acquisition. Whereas instance-based classification data reports values for every IMU reading at 0.1 second intervals, posture-based classification data represents the detection of the single instance that represents the beginning of the new posture. If additional reinforcement is provided due to false positives it may result in slower learning. If the reinforcement is provided too late after the posture change occurs the dog may not associate the reinforcement with the change in posture

²<http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>

and either no learning will occur or the dog will learn an association with a spurious behavior (sometimes referred to as superstitious behavior in the operant conditioning literature).

In an effort to reduce spurious sit posture classifications, the semi-autonomous system assumed that a dog could not revert from and to a sit posture in **less than 0.5 seconds**. For example, assume that in a one second interval at 10 Hz there are nine sit classifications and a single stand classification in the middle. The system will not assume the single stand **instance** classification (for 0.1 seconds) was an actual stand posture. The system requires 5 non-sit instance classifications (0.5 seconds) before assuming the dog **is no longer in the sit posture**. To meet the literature-inspired threshold of 0.5 seconds response latency for sit posture classification, whenever a dog is in a stand or eat posture, a single sit instance will immediately trigger a sit posture classification.

Therefore, in addition to timing, both instance- and posture-based classification accuracy has to be evaluated. Our evaluation of spurious sit posture classifications will be presented in Section 3.2.2. Instance-based accuracy uses the same 10-fold cross validation used in Section 3.1. Posture-based accuracy is measured in sensitivity (SE) and precision (PR). SE is the ratio of correct sit classifications, or True Positives (TP), over the number of TP and FN, $\frac{TP}{TP+FN}$. In other words, what percentage of the time the algorithm classified a sit relative to how often it should have. PR is the ratio of TP over the number of TP and false positive (FP), $\frac{TP}{TP+FP}$. In other words, what percentage of the time the algorithm classified a sit was it correct in doing so.

3.2.1. Setup

We recruited 16 **client-owned** dogs from the staff and community around the College of Veterinary Medicine at NC State. Because our emphasis was to compare the latency and accuracy of posture detection for a known posture, we were not concerned with dogs' **level of prior training**. See Table A.7 in the Appendix for information on the 16 dogs participating.

The experimental procedure consisted of three phases which we will describe in more detail below:

1. An *acclimation* phase allowed dogs to get comfortable with the apparatus and environment.
2. A *model building* phase produced labeled data for algorithm input.
3. An *evaluation* phase resulted in data for characterizing the performance of the algorithms.

Acclimation. Each dog was brought by the owner into a room specifically used for the experiment and the dog was allowed to roam freely on leash to acclimate to the new environment (see Figure 5). The room was configured such that the experimenters sat at a table that had a full view of an exercise pen (Xpen). The Xpen is a commercially available enclosure that is used to ensure the dog's safety while off leash. The Xpen was configured in a 6-foot by 6-foot square. Two video cameras were used to record the experiments. The owner was seated at a small table in an adjacent room and watched the experiment through a video display. The harness was placed on the dog, and plenty of food rewards were given to create a positive association with the harness.

Figure 5: The NC State Canine Facility where the second experiment took place. Shown on left is the Xpen in foreground, video recorder at top of the picture, and the table for experimenters. The dog owner area is in an adjacent room. Shown on right is a schematic of the facility.

During the acclimation process some of the dogs appeared uneasy with the Xpen and the unfamiliar surroundings. For all dogs before they entered the Xpen, the owner helped the VBT with the placement of the harness, coached the VBT on replicating the sit cue, and the VBT practiced accurately getting the dog to sit on cue. In this way if the dog didn't respond to the sit cue inside the Xpen, it was assumed the Xpen was causing the animal stress. If a dog was hesitant to enter the Xpen, a trail of treats and verbal encouragement from the VBT and owner were given as the dog entered the Xpen. For example, one dog walked gingerly on the concrete floor. The VBT identified the dog's anxiety and, after discussing with the owner, placed matting in the Xpen to provide the dog with more comfortable footing. For some dogs, the placement of the water bowl, location of the VBT, or the location of the owner were changed to make the dog more comfortable with the Xpen. Dog₁₆, a blind dog, circumnavigated the Xpen by exploring it with his nose before becoming comfortable within the space. Once the VBT and owners felt the dogs were acclimated to the Xpen, the model building process began.

Model Building. The owner's cue was used for the sit posture with the dog. After completing a sit, treats were dropped on the floor a few feet away from the dog so as to require the dog to stand up, walk, and eat the treats. Similar to Section 3.1.1, experimenters called out the posture changes, the sensor data was labeled, and the classifiers were built. Because there were labels for every instance, the instance-based accuracy results (presented below) are based off this model-building data, not the data collected during the evaluation phase.

Evaluation. After building the classifiers, the dogs were then brought into the Xpen to perform up to five minutes of sit, stand, and eat postures for each of two classifier experiments. The VBT had full authority to lessen the number of minutes and some of the dogs, due to age and physical ability, did not complete the two five minute sessions (see notes in Table A.7 in the Appendix). The VBT and everyone in the room were totally blind to which classification algorithm was used during the first five minute period. The base unit operator was the only person aware of which classification algorithm was used for the second five minute period.

The VBT was told the owner’s cue for the dog to sit and during two five minute periods the VBT gave the dog the sit cue while at the same time pressing a remote control’s cue button. The base unit recorded the sit cue along with the sensor data and the posture predictions. When the dog completed the sit, the VBT used the remote control’s mark button to record the mark and, optionally, a click sound was issued. The algorithm classified a sit posture if a sit instance-based classification was preceded by five non-sit instance-based classifications. The VBT’s mark and the algorithm’s posture classification were then compared to determine posture-based accuracy, and both were additionally compared against the three experimenters’ average time to determine latency.

3.2.2. Results

We analyzed the results of this experiment for response latency, accuracy, and outliers. Here we present these analyses.

Instance-based Accuracy. To evaluate the performance of the instance-based classification techniques we performed 10-fold cross validation over the labeled data created during the model-building phase of the experiment. As expected, the instance-based accuracy was extremely high.

Table 2 shows that for each dog with both classifications, RF classification had higher instance-based classification accuracy than VT. Only RF achieved 100% accuracy with Dog₀₈ while VT peak accuracy was 99.40%. Only Dog₀₁ had a VT accuracy greater than any of the RF classifications. Comparing in aggregate (last row of the table), the differences in accuracy were found to be significant (two-tailed t-test, $p=0.00001$).

Outlier analysis. Using the same data from Table 2, the percentage accuracy for both the VT and RF algorithms were checked for outliers using both the Grubbs’ and Dixons’ outlier tests. Grubbs’ test uses a sum-of-squares distance of the highest and lowest percentage accuracy. Dixons’ test uses an absolute distance of the highest and lowest percentage accuracy. These tests give a statistical analysis of whether any of the dogs had percentage accuracies outside the expected range of the group of dogs as a whole. These outliers would be dogs for whom the harness does not work as well as the others, dogs for whom the harness works better than the others, or dogs who for some other unknown reason did not perform as the others.

Using Grubb’s test, Dog₀₂ VT classification percentage accuracy is furthest from the rest of the VT classification percentage accuracy but not a significant outlier (two-sided, $Z=2.5457$ with a critical Z value of 2.8520, $p > 0.01$). Dog₀₇ RF classification percentage accuracy is furthest from the rest of the RF classification percentage accuracy but not a significant outlier either (two-sided, $Z=2.5262$ with a critical Z value of 2.8061, $p > 0.01$) using Grubbs’ test.

Using Dixon’s test for outliers at a 99% level of significance, Dog₀₇ VT classification accuracy (two-sided, $P=0.049$) and RF classification accuracy (two-sided, $P=0.216$) can be considered an outlier.

Given that Dog₀₇ is not the youngest, oldest, lightest, heaviest nor has any physical disabilities, we are reasonably confident in the harness’s usefulness for dogs of varying

Table 2: Confusion matrices and accuracy values for the 16 dogs in the second experiment.

Dog and Classification Algorithm		Sit Instances		Non-Sit Instances		Percentage Accuracy
		Sit	FP	Non-Sit	FN	
Dog ₀₁	VT	224	3	1098	5	99.40
	RF	228	0	1101	1	99.92
Dog ₀₂	VT	771	136	2075	166	90.41
	RF	929	8	2203	8	99.49
Dog ₀₃	VT	454	11	421	16	97.01
	RF	469	3	429	1	99.56
Dog ₀₄	VT	527	13	271	33	94.55
	RF	557	1	283	3	99.53
Dog ₀₅	VT	304	11	493	16	96.72
	RF	319	2	502	1	99.64
Dog ₀₆	VT	396	17	894	21	97.14
	RF	415	1	910	2	99.77
Dog ₀₇	VT	304	1	327	13	97.83
	RF	317	9	328	0	98.62
Dog ₀₈	VT	260	7	535	12	97.67
	RF	272	0	542	0	100.00
Dog ₀₉	VT	330	29	614	19	95.16
	RF	348	7	636	1	99.19
Dog ₁₀	VT	318	7	323	17	96.39
	RF	334	0	330	1	99.85
Dog ₁₁	VT	336	4	299	17	96.80
	RF	353	1	302	0	99.85
Dog ₁₂	VT	328	16	247	20	94.11
	RF	346	4	259	2	99.02
Dog ₁₃	VT	317	11	595	23	96.41
	RF	339	1	605	1	99.79
Dog ₁₄	VT	477	41	638	43	92.99
	RF	518	1	678	2	99.75
Dog ₁₅	VT	260	1	290	28	94.99
	RF	287	2	289	1	99.48
Dog ₁₆	VT	323	127	1132	19	96.94
	RF	340	4	1155	2	99.60
Mean	VT	370.6	27.2	640.8	29.3	95.906
	RF	398.2	2.8	659.5	1.6	99.566
StdDev	VT	134.4	42.0	475.2	37.6	2.160
	RF	166.3	2.9	501.1	1.9	0.361

Figure 6: Bar chart showing the mean time difference and one standard deviation for the 16 dogs for the RF and VT Classifiers, and Professional Trainer with respect to the three experimenters. Negative values for VT and Trainer indicate detection of sit prior to the labelers, and the narrower standard deviation for the VT algorithm shows a consistency improvement from the VBT.

age, weight, and physical ability. Of course, some undocumented characteristic of Dog₀₇ may be the cause of the outlier effect. Discovering the usefulness of the harness on a larger sample size at an animal shelter is an area of future work. But using the outlier tests on our 16 dogs gives us some confidence that the harness works on dogs with varying age, weight, and physical ability.

Timing. To evaluate the timing of the algorithms, we compared the time the algorithm produced a posture-based classification of a sit to the time the VBT marked the sit, and to the average time of the three experimenters. Figure 6 summarizes the results of the posture-based timing analysis. There are a few interesting things to note about that Figure. First, RF posture-based classification performed worse than the VT classification. We will explore this in more detail below. Second, the mean latency of the VT algorithm and the VBT was actually negative. While this may seem strange at face value, it can be explained by two things: 1) The three experimenters were consistently slower than the VBT and algorithm; and 2) Because entering a sit posture is a process, the determination of when the sit begins is actually subjective. One implication of this second explanation is that the VT algorithm actually learns to identify the process of entering a sit, which enables it to have excellent timing. The third interesting thing to note about that Figure is that the mean time of the VT algorithm is slightly lower than the VBT, and the variance is much lower.

Table 3 shows the details of the same data summarized in Figure 6. It contains the mean latency and standard deviation for both the VBT and the classifier. For 12 out of the 16 dogs (75%) RF posture-based classification was above the literature-inspired threshold of 0.5 seconds response latency. According to a two-tailed t-test, the observed differences between the latency of VT and RF was significant ($p=0.0002$).

The VBT had only one trial, Dog₁₂ VT classification, that exceeded the 0.5 second response latency goal with a latency of 0.772 seconds. For that experiment, the dog did

Table 3: The mean and standard deviation response latency for the RF and VT algorithms and VBT in comparison to the mean of three experimenters. Note that the entry for the VBT Δt indicates the VBT’s timing relative to the mean of the three experiments during the evaluation period where the algorithm indicated in that row was also being evaluated. To compare the VBT’s timing to the algorithm’s, compare the values in the columns of that row.

Dogs and Algorithm		Trainer Δt		Classifier Δt	
		Mean	StdDev	Mean	StdDev
Dog ₀₁	VT	0.035	0.209	-0.357	0.176
	RF	-0.136	0.227	7.506	3.589
Dog ₀₂	VT	-0.319	0.246	-0.242	0.188
	RF	-0.364	0.420	2.036	4.024
Dog ₀₃	VT	-0.174	0.257	-0.075	0.283
	RF	-0.179	0.374	1.643	3.436
Dog ₀₄	VT	-0.154	0.188	0.058	0.169
	RF	-0.641	0.938	0.448	5.497
Dog ₀₅	VT	-0.120	0.141	0.079	0.425
	RF	-0.227	0.577	1.760	2.551
Dog ₀₆	VT	-0.126	0.166	0.139	0.171
	RF	1.631	1.484	3.113	2.522
Dog ₀₇	VT	-0.145	0.196	-0.176	1.660
	RF	-0.109	0.179	0.737	2.488
Dog ₀₈	VT	0.234	1.558	0.200	1.614
	RF	-0.405	1.302	2.246	3.386
Dog ₀₉	VT	-0.276	0.690	-0.449	1.534
	RF	-0.240	0.651	1.684	2.039
Dog ₁₀	VT	0.000	0.284	0.070	0.380
	RF	-0.067	0.158	1.070	2.170
Dog ₁₁	VT	-0.045	0.260	-0.107	0.109
	RF	-1.154	2.833	-0.560	4.821
Dog ₁₂	VT	0.772	1.024	1.030	1.059
	RF	-1.149	1.307	0.467	3.186
Dog ₁₃	VT	-0.106	0.104	0.126	0.151
	RF	-0.103	0.126	2.837	4.843
Dog ₁₄	VT	-2.530	1.873	-2.140	1.852
	RF	-0.038	0.106	1.115	3.114
Dog ₁₅	VT	0.080	0.157	0.080	0.154
	RF	0.104	0.188	2.388	1.758
Dog ₁₆	VT	0.032	0.593	0.474	3.63
	RF	-0.092	0.141	5.494	4.565

Figure 7: Posture-based classification sensitivity, $TP \div (TP + FN)$.

the second most number of sits, 46 sits during the five minute period and the remote control marking device had some failures marking labels.

Posture-based Accuracy. The base unit determined posture changes by using instance-based classifications. For all algorithms, whenever the previous five instance classifications (0.5 seconds) were classified as non-sits and a single sit instance classification occurred, the algorithm classified a sit posture. Timing comparisons for when the VBT and the algorithm determined a treat should be given were made using experimenters as a point of comparison.

A sit posture classified by the algorithm when the experimenters determined a sit had not occurred was considered a FP. A sit posture that was not classified when the experimenters determined one had occurred was considered a FN. A sit posture classified correctly when the experimenters determined one had occurred was considered a True Positive (TP).

Table 4 shows the complete posture accuracy information for all of the dogs participating. Figure 7 shows the SE for the VT and RF algorithms. The VT algorithm was designed to limit FN and the sensitivity was 100 for all dogs. The VT algorithm had no FN's for the 16 dogs. Differences in PR for the RF and VT classifiers were significant (two-tailed t-test, $P=0.000108$). Note that a t-test cannot be used for SE VT and RF comparison due to the VT StdDev=0.

Figure 8 shows the PR for the VT and RF algorithms. Although there were FP for the VT classification, only for Dog₀₉ did RF SE exceed VT SE.

RF Classification Elapsed Time. Timing analysis was done to determine why the instance classification of the RF algorithm consistently exceeded the VT algorithm but for the posture classifications the VT algorithm consistently exceeded RF. If the RF algorithm was more accurate but required a longer time to classify, then the RF posture classifications might fall behind the 10Hz data collection rate.

Table 4: The number of sits, FP, FN, PR, and SE for each of the 16 dogs in the timing experiments.

Dogs and Algorithm		Sits	FP	FN	PR $\times$ 100	SE $\times$ 100
Dog ₀₁	VT	8	6	0	57.14	100.00
	RF	11	6	0	64.71	100.00
Dog ₀₂	VT	29	0	0	100.00	100.00
	RF	20	27	2	42.55	90.91
Dog ₀₃	VT	17	4	0	80.95	100.00
	RF	19	46	2	29.23	90.48
Dog ₀₄	VT	34	3	0	91.89	100.00
	RF	31	9	3	77.50	91.18
Dog ₀₅	VT	35	0	0	100.00	100.00
	RF	33	24	6	57.89	84.62
Dog ₀₆	VT	46	1	0	97.87	100.00
	RF	39	11	8	78.00	82.98
Dog ₀₇	VT	49	1	0	98.00	100.00
	RF	42	14	4	75.00	91.30
Dog ₀₈	VT	40	5	0	88.89	100.00
	RF	32	5	6	86.49	84.21
Dog ₀₉	VT	27	13	0	67.50	100.00
	RF	37	15	4	71.15	90.24
Dog ₁₀	VT	35	0	0	100.00	100.00
	RF	37	16	7	69.81	84.09
Dog ₁₁	VT	5	0	0	100.00	100.00
	RF	18	3	3	85.71	85.71
Dog ₁₂	VT	46	2	0	95.83	100.00
	RF	46	34	10	57.50	82.14
Dog ₁₃	VT	16	3	0	84.21	100.00
	RF	13	4	4	76.47	76.47
Dog ₁₄	VT	30	1	0	96.77	100.00
	RF	23	9	6	71.88	79.31
Dog ₁₅	VT	26	0	0	100.00	100.00
	RF	37	12	7	75.51	84.09
Dog ₁₆	VT	20	2	0	90.91	100.00
	RF	20	6	2	76.92	90.91
Mean	VT	28.9	2.6	0.0	90.6	100.0
	RF	28.6	15.1	4.6	68.5	86.8
StdDev	VT	13.2	3.4	0.0	12.7	0.0
	RF	10.9	12.0	2.7	15.3	5.8

Figure 8: Posture-based classification precision, $TP \div (TP + FP)$.

Calculating the runtime for RF classification showed on average 0.124 seconds per classification—well above the 0.1 second interval between sensor readings. This compares to the 0.001 second average runtime for each VT classification of the sensor information. VT classification which simply checked if two sensor values were within a predefined range was approximately 100 times faster than RF classification. With the sensor receiving data at 10Hz, the RF classifier was generally falling behind and classifying instances in the past.

To verify the latency problem of RF, data were replayed through the classifier allowing the classifier time to finish. The sit, FP, FN, PR, and SE results of posture accuracy allowing RF more time as compared to VT is shown in Table 5. Table 5 uses the classifier name RF* for the same Random Forest classifier replayed to allowed the classifier to run 0.125 seconds before the next data instance was classified. Comparing Table 3 and Table 5, the number of FN went from more than 70 with RF classification to only 6 FN for RF*. The observed difference between the PR of the RF and VT algorithms was not found to be significant in this case (two-tailed t-test, $P=0.4249$). The implications of this on the design of a computer-based training system are somewhat unclear at this point. Collecting data at 8Hz rather than 10Hz will naturally produce higher response latencies; however, it may also yield improvements in SE and PR. **Future studies will investigate** trade-offs between data collection rate, response latency, and accuracy.

RF Classification requiring two sit instances. We did an analysis to understand why RF PR was generally less than VT PR. The difference between PR for RF and VT could be due to the distribution of instance-based FP. Many instance-based FP clustered together might result in a single posture-based FP, whereas many instance-based FP dispersed could result in multiple posture-based FP.

Figure 9 shows that whether VT sit posture classification requires one sit (1Sit) or requires two consecutive sit instance classifications (2Sit), there was no effect on SE because in both classifications for all dogs, SE was 100. SE for RF improved for all dogs when using 2Sit.

Table 5: A comparison of the VT algorithm and RF*, replaying the data 0.125 second intervals (8Hz).

Dogs and Algorithm		Sits	FP	FN	PR $\times$ 100	SE $\times$ 100
Dog ₀₁	VT	8	6	0	57.14	100.00
	RF*	8	4	0	66.67	100.00
Dog ₀₂	VT	29	0	0	100.00	100.00
	RF*	29	67	0	30.21	100.00
Dog ₀₃	VT	17	4	0	80.95	100.00
	RF*	17	13	0	56.67	100.00
Dog ₀₄	VT	34	3	0	91.89	100.00
	RF*	34	11	0	75.56	100.00
Dog ₀₅	VT	35	0	0	100.00	100.00
	RF*	35	33	1	51.47	97.22
Dog ₀₆	VT	46	1	0	97.87	100.00
	RF*	46	5	1	90.20	97.87
Dog ₀₇	VT	49	1	0	98.00	100.00
	RF*	49	6	0	89.09	100.00
Dog ₀₈	VT	40	5	0	88.89	100.00
	RF*	40	30	2	57.14	95.24
Dog ₀₉	VT	27	13	0	67.50	100.00
	RF*	27	20	0	57.45	100.00
Dog ₁₀	VT	35	0	0	100.00	100.00
	RF*	35	6	1	85.37	97.22
Dog ₁₁	VT	5	0	0	100.00	100.00
	RF*	5	0	0	100.00	100.00
Dog ₁₂	VT	46	2	0	95.83	100.00
	RF*	46	17	0	73.02	100.00
Dog ₁₃	VT	16	3	0	84.21	100.00
	RF*	16	1	0	94.12	100.00
Dog ₁₄	VT	30	1	0	96.77	100.00
	RF*	30	8	0	78.95	100.00
Dog ₁₅	VT	26	0	0	100.00	100.00
	RF*	26	1	0	96.30	100.00
Dog ₁₆	VT	20	2	0	90.91	100.00
	RF*	20	11	1	64.52	95.24
Mean	VT	28.9	2.6	0.0	90.6	100.0
	RF*	28.9	14.6	0.4	72.9	98.9
StdDev	VT	13.2	3.4	0.0	12.7	0.0
	RF*	13.2	17.1	0.6	19.4	1.8

Figure 9: Posture-based classification sensitivity improvement by requiring two consecutive sit instances for sit posture classification sensitivity. Note that there was no improvement in VT classification sensitivity since it was already at 100. The improvement was with respect to RF* (with data replayed at 8Hz).

Figure 10: Example time line of instances where a 2Sit classification would reduce FNs.

Given that SE can only improve if the number of FN decreases, an example of how requiring 2Sit reduces FNs is shown in Figure 10. Requiring only one sit instance for a sit posture would produce a FP at t_{11} and since the system assumes the dog could not revert the sit posture in under a half second (see Section 3.2), the sit instances after t_{14} would be assumed to be a continuation of the sit at t_{11} and would not be classified as a sit posture. Requiring two sit instances not only removes the FN at t_{11} but also reduces a FP at t_{15} . In general, with dispersed spurious sit instance classifications, requiring two sit instances not only reduces FPs but also reduces FNs.

Figure 11 shows that changing from VT 1Sit to 2Sit had little effect on PR. Only Dog₁₃ showed an increase in PR, going from 84.21 1Sit to 88.89 for 2Sit. For RF 2Sit, only Dog₀₈ and Dog₀₉ had a reduction in PR. Dog₀₈ went from 1Sit RF 86.49 PR to 2Sit RF 65.57 PR. Dog₀₉ went from 1Sit RF 71.15 PR to 2Sit RF 67.50 PR.

Since requiring two consecutive sit instance classifications for a sit posture classification generally has a positive effect on PR and SE, it seems that RF FP are more dispersed and VT FP are more clustered. It is a topic for future work to investigate the effects of 2Sit posture-based classification has on how dogs learn from our computer-assisted training system.

3.3. Training

The third experiment further validates the effectiveness of the system by showing the conditions under which it successfully is able to semi-autonomously **train** a dog

Figure 11: Posture-based classification PR improvement by requiring two consecutive sit instances for sit posture classification. The improvement was with respect to RF* (with data replayed at 8Hz).

to sit for food rewards. Evidence of a dog’s learning is shown in a learning curve of cumulative sits over time.

This experiment, although presented last, actually occurred before the other two chronologically, and its results motivated the need to conduct the other two experiments. We have chosen to present it last because it provides further validation of the importance of timing, and demonstrates evidence of dogs learning from our computer-assisted system. In contrast to the other two experiments, the goal in presenting the data generated from this experiment is not to demonstrate how fast and accurate our algorithms can be, but to show that when they are fast and accurate and are also responsible for providing feedback to the dogs there is some evidence of learning occurring. We further note that the sample size for this experiment is too small to draw any generalizable conclusions. But, we hope the data are informative of the potential of the system we present in this paper.

Because this experiment occurred earlier some of the algorithms used were slightly different from those presented in the above sections. Training was done using two classification algorithms, RF and a Threshold Classifier, which was an earlier version of VT that used fixed-sigma multiples and a disjunction of sensor value expressions (rather than the conjunction used in VT). The RF algorithm used is the same algorithm used in the previous experiments. Additionally, we used a slightly more complex Univariate Spline Smoothing algorithm³ to filter the data, rather than the simpler moving average filter used above.

Three dogs, Dog_α, Dog_γ, and Dog_δ, were used. No professional trainer was present, instead the system used during the experiment was semi-autonomous as it still required the input of human labelers for model building. The automated system validated the understanding of the latency effect on the dog’s learning of the desired

³<http://docs.scipy.org/doc/scipy-0.15.1/reference/generated/scipy.interpolate.UnivariateSpline.html>

posture.

3.3.1. Setup

The procedures involved were similar to the previous experiments. Each dog was brought into a room specifically used for the experiment and the dog was allowed to roam freely to acclimate to the new environment. We did not use experimenters to provide a point of comparison to when a sit occurred. Instead, two video cameras were used to record the experiment and later analysis determined when a sit occurred for algorithm timing comparisons. The Xpen used was configured slightly larger, taking up a 15-foot by 6-foot area of the room. The larger Xpen was used because two dispensers alternatively dispensed treats requiring the dog to stand and walk between sits.

The harness was placed on the dog similar to the previous experiment. One difference was that after the classifiers were built, we performed a final check of the classifiers to ensure their accuracy. Any technical errors could affect the reinforcement accuracy and rate, factors that influence training. This avoided confusing the dogs by having treats dispensed at seemingly random times. This informal validation of the model consisted of repeating the same sequence of postures two to four times, and visually comparing the labels output by the classification algorithm.

In addition, another exception was that the dogs were introduced to the computer-controlled treat dispenser during and after the model building phase. The dispensers were set to make a beep immediately upon being triggered, and prior to dispensing a treat. In learning theory, the beep serves as a secondary reinforcer, which is then followed by the primary reinforcer, the treat. To acclimate the dogs, one dispenser was fired 10 times, followed by the other across the Xpen 10 times, and finally both were alternated a total of 10 times more. This process allowed the dog to establish a mental association between the sound of the beep and the forthcoming treat. It also helped to accustom each dog to move between dispensers, which we hoped would help to prompt each dog to get up and move, thereby readying the dog for the next sit.

Once the dog was acclimated to the treat dispensers, it was placed in the Xpen. An experimenter was also in the Xpen to avoid confusion and stress on the dogs. If the dogs appeared confused or stressed at the beginning of the experiment, the experimenter would use body language to encourage the dogs to reposition and sit.

Timing information was extracted after the trials were complete. Three raters coded the videos for the timing of sit behaviors. The exact time of each sit behavior was defined by the first frame that shows the canine's upper thigh making contact with the ground. We averaged the times of all three rates. The rates also annotated false positives and negatives.

3.3.2. Results

Table 6 summarizes the results of the algorithm performance during the third experiment.

Timing and Accuracy. As can be seen in Table 6 and as consistent with the results in Section 3.2, RF posture-based classification performed worse than the VT algorithm. Figure 12 further illustrates how the mean latency of the RF classifier was above the

Table 6: The number of sits, FP, FN, PR, and SE for each of the three dogs in the training experiments. The mean and standard deviation response latency for the RF and VT algorithms in comparison to the mean of the three experimenter video analysis.

Dogs and Algorithm	Sits	FP	FN	PR $\times$ 100	SE $\times$ 100	Video Analysis Δt		Duration
						Mean	StdDev	
Dog $_{\alpha}$ VT	46	0	0	100.00	100.00	0.290	0.251	15:18
Dog $_{\delta}$ VT	23	6	0	79.31	100.00	-0.141	0.108	17:20
Dog $_{\gamma}$ RF	16	1	8	94.12	66.67	1.233	2.398	18:31

Figure 12: Bar chart showing the mean time difference and one standard deviation for the 3 dogs for the RF and VT Classifiers with respect to the video analysis.

Figure 13: The cumulative number of sits offered by each dog over time. Series labels include the mean and standard deviation response latency.

literature-inspired threshold of 0.5 seconds and the standard deviation of the VT classifier was also much less than that of the RF classifier. Note that the timing results reported here differ from those from experiment two in that they were calculated based on later video analysis, rather than based on a point of comparison identified in real-time by experimenters. This explains why the VT algorithm had a negative latency in the second experiment, and a slightly positive one in the third.

In considering the evidence of learning below, keep in mind that: for Dog_α, the VT algorithm was exceedingly accurate and very timely; for Dog_δ, the VT algorithm was pretty accurate and exceedingly timely; and for Dog_γ, the RF algorithm with neither especially accurate nor timely.

Evidence of Learning. Beyond timing, the analysis of our semi-autonomous system’s performance was in the form of characterizing the dogs’ postures. Ideally the dogs would have rapidly learned to sit, receive a treat (which required the dog to stand with a lowered head to eat), and repeat immediately.

Figure 13 also illustrates the rate at which each of the dogs offered sits. Ideally, we would see super-linear growth in these data; however, faster linear growth was better than slower linear growth as it indicates the dog was more enthusiastically offering the desired posture. At the beginning of the semi-autonomous experiment, when the dogs appeared confused or stressed the experimenter used body language to encourage the dog to the sit. Figure 13 also indicates, for the trials where the experimenter was involved and when he stopped giving body language feedback to the dog. Note that the data in Figure 13 seems to indicate that Dog_γ performed similarly to Dog_δ; however, what wasn’t represented in the Figure was that the dog appeared disinterested, laying

Figure 14: Orientation of Dog_α to experimenter vs. treat dispenser grouped by thirds of the experimental trial.

down by the time the high-latency reinforcement occurred, and required encouragement from the experimenter to get up to receive the treat (despite quickly retrieving treats early in the trial).

For Dog_α it was very noticeable that the dog had learned that the sit posture would be reinforced and **was highly motivated to seek reinforcement**, since he performed more than **twice** the number of sits with respect to the other dogs in the same period of time. One of the interesting things observed for both Dog_α and Dog_δ was that initially they tended to sit while orienting towards the experimenter who was located in the corner of the room. Over the course of the trial, both dogs started to ignore making eye contact with the experimenter in favor of focusing on the treat dispensers. Figure 14 shows how Dog_α started to focus less on the experimenter, and instead focus on the treat dispensers over time. It is believed that this was indicative of the dog realizing that the experimenter was not in control of the treats, and subsequently lost interest in him. This is strong evidence that the dog was learning from the training system with little influence by the experimenter.

4. Conclusion and Future Work

In this paper we described, what is to our knowledge, the first semi-autonomous canine training system capable of capturing and reinforcing postures. We briefly discussed the hardware platform for our research, as well as the central ergonomic and performance design considerations. Putting the dog at the center of our design motivations, these considerations have yielded a comfortable, wearable system and algorithms

for high-accuracy and low-latency recognition of postural changes from multiple inertial measurement units. We presented a simple variance-based threshold classifier that identifies the means and standard deviation of each of the six IMU axes, identifies boundaries between those values that differentiate the data associated with different postures, and construct a set of decision rules accordingly. We reported on three experiments, the first experiment illustrates how response latency and noise sensitivity must be managed effectively through data pre-processing to achieve a balance between timely and accurate feedback. The second experiment includes a more extensive validation of the timing experiments using a professional dog trainer as a point of comparison. Finally, the third experiment is intended to characterize how the system can promote learning when accuracy and latency goals are achieved.

Results of our first experiment with two dogs indicate that using a moving average filter with a window size of four, for our system, provides the smallest number of false negatives. We note that this result is likely specific to our hardware setup and algorithm choice, and encourage the reader to pay more attention to the process of validating the data pre-processing than the specific outcome of the first experiment.

The results of the second experiment indicate that the VT algorithm is demonstrably superior to the RF algorithm in terms of posture-based accuracy and response latency, although the RF algorithm does have better instance-based accuracy. Further analysis suggested one plausible explanation for this result is that the computation time required for the RF algorithm exceeded the time interval between IMU samples, causing a chronic lag and reduction in posture-based accuracy and increase in latency. In addition, results of the second experiment indicated the VT algorithm had performance favorably-comparable to a professional trainer. The mean time of the VT algorithm was slightly lower than the professional trainer, and the variance was much lower.

The qualitative results of the third experiment using three dogs showed faster linear growth in the sit rate when the algorithm was both fast and accurate, indicating that the dog receiving that algorithmically-controlled feedback was more enthusiastically offering the desired posture. We also observed that over time, the dogs started to focus less on the experimenter, and instead focus on the treat dispensers, which can be interpreted as further evidence of learning. While the results of the third experiment are exceedingly encouraging, especially when taken in combination with the results of the second experiment, we acknowledge that the sample size is too low to draw any generalizable conclusion at this point in time.

Despite a small sample size and the resulting lack of external validity, there are some exciting implications for ACI work that results from our findings. First, we believe our results support the conclusion that a mixed-initiative system combining human interaction with algorithmic interpretation can lead to successful animal training. By showing that our algorithm has accuracy and timing that compares favorably to a professional trainer, we demonstrated that a computer can be an effective collaborator in animal training. This paves the way to future mixed-initiative ACI systems. Second, we have verified that similar timing constraints that were identified with human-animal interactions are relevant to animal-computer interactions. We have shown how important the influence of timing can be on dogs' abilities to learn from computationally-controlled feedback. Future work in ACI should take this finding into account, and make sure to have accurate and timely feedback to animals as a primary design goal.

Building on our success to this point, we have identified several areas for continued development and future research. First, we’re interested in streamlining and further optimizing the sensor platform for its form and function. Continued improvements in the size and bulkiness of the harness are needed. While all dogs in our experiments acclimated to the harness in a matter of minutes, some dogs with contact sensitivity may still find it inhibiting to wear. Also, it may not be suitable for significantly smaller dogs. We plan to develop a miniaturized version of the system that is more comfortable for all dogs, and also is suitable for small dogs.

As mentioned earlier, it remains an open question whether collecting IMU data at 10Hz produces better results than another rate, like 5Hz or 20Hz for example. The answer may be that 10Hz with an efficient algorithm like VT is sufficient, but that remains to be investigated. Further, as we’ve evaluated algorithms for accuracy on multiple postures in prior work, we’re interested in replicating the timing results across multiple postures as well. We strongly believe that any single posture in a one-vs-other classification scheme will result in similar timing results; however, things may be less straightforward when using our system for detecting sit, stand, eat, and down at the same time. In particular, the VT algorithm will certainly take longer to run as it has more thresholds to check in that case. Again, it remains an open question if these results will generalize to multiple posture detection. We are also, of course, interested in conducting a training experiment with a larger sample size to formally validate that our system, when accurate and timely, produces behavior change in dogs.

Looking farther down the road, we are also interested in having the ability to request specific behaviors from the dogs using a “discriminative stimulus” [?]. This would entail the pairing of a unique stimulus with each behavior to serve as a “cue.” The process of putting a behavior under stimulus control is well documented among trainers, but operationalizing this process to a semi-autonomous computer system presents additional challenges.

We are also interested in taking our semi-autonomous system and making it fully autonomous. We’ve referred to the system as computer-assisted because it currently requires human assistance to provide labels for supervised classification of postures. We are currently investigating ways to build an unsupervised posture detection system and have had some early success in that regard [?]. The system uses pre-specified declarative expert knowledge to automatically detect a given set of postures, but accuracy is not yet approaching that of our existing techniques.

Lastly, we hope that the successful iterative user-centered approach based on the criteria of canine comfort and well-being followed in the design of this computer-assisted training system provides enriched canine stimulation, aids canine training, and may serve as a model for the development of other ACI technologies involving canine wearable sensors and remote delivery of primary reinforcers.

In recent years there has been a rapid proliferation of Internet-of-Things (IoT)-enabled devices in markets that are positively affecting many sectors of the global economy: retail, healthcare, transportation, and energy. These technologies are also finding their way into consumer pet products. We sincerely hope that efforts like ours will serve as an example of design process and criteria that could be followed to ensure this new space of technologies appearing in the consumer market are providing value not only for humans but for animals as well.

5. Acknowledgments

We would like to thank Shatorupa Ghosh and Elizabeth Furr for technical assistance during the 16 dog experiments and all the dog owners for their participation.

This work is supported by the US National Science Foundation under the Cyber Physical Systems Program (IIS-1329738).

6. Vitae

To be completed.

References

Appendix A. Dog Demographics

Table A.7: Dog Demographics

Dog	Sex	Age	Breed	Owner reported weight	Any notes
Dog _{α}	Male	3 years	Labrador Retriever	31 kg	Only used for training
Dog _{β}	Male	2 years 9 months	Labrador Retriever	32 kg	Only used for training
Dog _{γ}	Male	3 years 5 months	Weimaraner	34 kg	Only used for training
Dog _{δ}	Male	3 years 2 months	Retriever Mix	21 kg	Only used for training
Dog ₀₁	Male	8 years 5 months	Otterhound	38 kg	Did sits very slowly
Dog ₀₂	Male	1 year 2 months	Collie	21 kg	
Dog ₀₃	Female	11 months	Poodle	17 kg	
Dog ₀₄	Female	3 years 2 months	American Staffordshire Terrier	22 kg	
Dog ₀₅	Male	6 years 1 month	Labrador Retriever	19 kg	
Dog ₀₆	Male	6 years 1 month	Border Collie	24 kg	
Dog ₀₇	Female	4 years 6 months	Portuguese Water Dog	17 kg	
Dog ₀₈	Male	4 years 7 months	American Staffordshire Terrier	20 kg	Three legged
Dog ₀₉	Male	1 year 11 months	German Shepherd	32 kg	Had odd, sliding sit
Dog ₁₀	Male	3 years 9 months	Australian Shepherd	22 kg	
Dog ₁₁	Female	7 years 8 months	Labrador Retriever	32 kg	Short experiment due to Osteochondritis Dissecans (OCD)
Dog ₁₂	Female	11 years 8 months	Border Collie	18 kg	
Dog ₁₃	Female	2 years 3 months	Staffordshire Bull Terrier	33 kg	
Dog ₁₄	Male	10 years	Australian Shepherd	26 kg	
Dog ₁₅	Male	3 years 5 months	Labrador Retriever	32 kg	Battery fell off during experiment
Dog ₁₆	Male	4 years 8 months	Beagle	12 kg	Blind

Figure A.15: Left to right Dog_α and Dog_β .

Figure A.16: Left to right Dog_γ and Dog_δ .

Figure A.17: Left to right Dog_{01} and Dog_{02} .

Figure A.18: Left to right Dog₀₃ and Dog₀₄.

Figure A.19: Left to right Dog₀₅ and Dog₀₆.

Figure A.20: Left to right Dog₀₇ and Dog₀₈.

Figure A.21: Left to right Dog₀₉ and Dog₁₀.

Figure A.22: Left to right Dog₁₁ and Dog₁₂.

Figure A.23: Left to right Dog₁₃ and Dog₁₄.

Figure A.24: Left to right Dog₁₅ and Dog₁₆.