Article

Gradient-Based Multi-Objective Feature Selection for Gait Mode Recognition of Transfemoral Amputees

Gholamreza Khademi [1,†,‡], Hanieh Mohammadi [1,‡] and Dan Simon [1,*]

[1] Department of Electrical Engineering and Computer Science, Cleveland State University, Cleveland, Ohio, USA; d.j.simon@csuohio.edu

* Correspondence: g.khademi17@csuohio.edu; Tel.: +1-216-801-9346

† Current address: 2121 Euclid Avenue, Cleveland, Ohio, USA, 44115

‡ These authors contributed equally to this work.

Abstract: One control challenge in prosthetic legs is seamless transition from one gait mode to another. User intent recognition (UIR) is a high-level controller that tells a low-level controller to switch to the identified activity mode, depending on the user's intent and environment. We propose a new framework to design an optimal UIR system with simultaneous maximum performance and parsimony for gait mode recognition. We use multi-objective optimization (MOO) to find an optimal feature subset that creates a trade-off between these two conflicting objectives. The main contribution of this paper is two-fold: (1) a new gradient-based multi-objective feature selection (GMOFS) method for optimal UIR design; and (2) the application of advanced evolutionary MOO methods for UIR. GMOFS is an embedded method that simultaneously performs feature selection and classification by incorporating an elastic net in multilayer perceptron neural network training. Experimental data are collected from six subjects, including three able-bodied subjects and three transfemoral amputees. We implement GMOFS and four variants of multi-objective biogeography-based optimization (MOBBO) for optimal feature subset selection, and we compare their performances using normalized hypervolume and relative coverage. GMOFS demonstrates competitive performance compared to the four MOBBO methods. We achieve a mean classification accuracy of 97.14% ± 1.51% and 98.45% ± 1.22% with the optimal selected subset for able-bodied and amputee subjects, respectively, while using only 23% of the available features. Results thus indicate the potential of advanced optimization methods to simultaneously achieve accurate, reliable, and compact UIR for locomotion mode detection of lower-limb amputees with prostheses.

Keywords: User intent recognition; transfemoral prosthesis; multi-objective optimization; biogeography-based optimization

1. Introduction

Prosthetic legs have significantly enhanced the lifestyle of individuals with a transfemoral amputation. Prostheses help lower-limb amputees regain their walking mobility for activities such as level walking, stair ascent and descent, incline walking, sitting and standing, etc. One active research area is the development of a functional control system for each walking task [1–4]. The main design objective is to enable amputees to achieve walking that is similar to that of able-bodied persons, while minimizing metabolic energy expenditure. Challenges include recognizing gait modes automatically, selecting the appropriate control system corresponding to the identified gait mode, and achieving a smooth transition in real time. Activity mode recognition must be achieved in parallel with control system development to address these problems. Activity mode recognition is referred to as high-level control, while control system design for each walking activity is referred to as low-level control [5]. The focus of this paper is the development of a high-level control system.

In the design of an intent recognition system, several questions arise, including which input signals and machine learning algorithms will provide a UIR system with fast and reliable prediction performance. Previous research has addressed these questions in different ways. For instance, surface electromyography (sEMG) signals were used to train UIR [6,7]. Although sEMG resulted in high classification accuracy, [8] reported uncertain performance due to sEMG signal variability in real-world conditions. Variation could be because of electrode shift [9], skin temperature change [10], or muscle volume change [11]. Therefore, external sensors on the prosthesis have received significant attention. For instance, classifiers have been trained with data collected from mechanical sensors [12,13], optical distance sensors [14], and inertial measurement units [8]. In addition, [15,16] showed that the fusion of sensory measurements could enhance learning, although the amputee subject could be inconvenienced by wearing additional sensors. Various supervised machine learning algorithms have been implemented to build UIR systems, including linear discriminant analysis (LDA) [17], quadratic discriminant analysis (QDA) [18], Gaussian mixture models (GMMs) [12], support vector machines (SVMs) [16], and artificial neural networks (ANNs) [6]. To avoid the need for user-specific classifier training, [19] proposed a user-independent UIR system in which classifier performance is robust to user-specific characteristics.

Current UIR systems have been designed with one goal in mind: highest possible prediction accuracy. In clinical applications, it is extremely important that UIR can accurately predict activity modes with substantially different characteristics, because misclassification can cause a loss of balance [8,20]. However, there remains a gap in the design of UIR with parsimony, or low complexity. A UIR has low complexity if it can be trained with only significant features extracted from minimal sensing hardware. UIR with low complexity is important because such systems: (1) eliminate unneeded body-worn sensors that may be irritating and cumbersome; (2) avoid numerical instability and overfitting during training; (3) are robust to noisy measurement signals and sensor failures; and (4) decrease computational effort, which is important for real-time operation. These reasons motivate us to develop a new framework for UIR that simultaneously achieves maximum accuracy and maximum parsimony. Parsimony and accuracy are two conflicting objectives. This paper is the first attempt to find a compromise solution for this problem.

The main contributions of this paper are two-fold: (1) a new MOO method called GMOFS for optimal feature subset selection; and (2) the application of four evolutionary MOBBO methods for the UIR problem, including vector evaluated BBO (VEBBO), non-dominated sorting BBO (NSBBO), niched Pareto BBO (NPBBO), and strength Pareto BBO (SPBBO). We have chosen to use BBO in this paper as the evolutionary algorithm (EA) because of its demonstrated effectiveness and recent popularity for optimizing real-world problems [21,22]. MOBBO methods have the potential to find the global optimum [23,24]; however, they are computationally expensive due to the many required fitness function evaluations. To avoid this drawback, we propose GMOFS for feature selection.

Several different types of feature selection methods have been proposed. *Filter* methods are feature selection methods that assess the quality of a subset of features independently or with respect to the output class [25]. *Wrapper* methods are feature selection methods that assess the quality of a subset of features by measuring the prediction accuracy of a classifier that is trained with that subset [26]. *Embedded* methods are feature selection methods that overcome the disadvantages of filter and wrapper methods. Unlike filter methods, embedded methods account for the bias of the classifier, and unlike wrapper methods, they are computationally efficient [27,28]. Various embedded feature selection algorithms have been proposed, mostly for linear problems with a single objective [28,29]. Embedded methods also incorporate regularization algorithms, such as ridge regression [30], least absolute shrinkage and selection [31], and elastic nets [32].

GMOFS is our newly proposed embedded method that simultaneously performs feature selection and classification, and that accounts for multiple objectives in nonlinear systems such as UIR. GMOFS incorporates an elastic net in multilayer perceptron (MLP) neural network training. The elastic net uses a Lagrange multiplier with a complexity parameter to reduce the feature set to an optimal subset, and the MLP classifier is trained with the optimal subset. We investigate the influence of the complexity parameter on the solution of the constrained MLP optimization problem. We then use the optimization solutions to obtain a GMOFS Pareto front, which is a set of non-dominated solutions that are equally important apart from the designer's subjective preference of objectives.

Section 2 presents a general framework for UIR. In Section 2.1, a rich set of signals reflecting various walking tasks are collected experimentally from three able-bodied and three amputee subjects. In addition, the data are filtered and processed to eliminate noise and missing data points. In Section 2.2, we use both disjoint windowing and overlapped windowing to extract data frames. The length of the data frame and the increment of the moving window are chosen to compromise the richness of the data and the computational effort, while taking real-time computational constraints into account. In Section 2.3, various time-domain (TD) and frequency-domain (FD) features are extracted from each data frame for each measurement signal. A training data set is obtained in which all features are normalized to have a zero mean and unity variance. In Section 2.4, we use a pre-selection approach to exclude insignificant features, and then apply MOO for final feature selection. We implement GMOFS and four variants of MOBBO to minimize the size of the selected feature subset and maximize the prediction accuracy. In Section 2.5, the performance of several classifiers, including LDA, QDA, SVMs with both linear and RBF kernels, MLPs, and decision trees (DTs), are compared, and the best one is selected for UIR. In Section 2.6, MVF is implemented to avoid sudden jumps between identified classes and enhance UIR performance. Section 3 discusses the experimental setup and classification results for the optimally designed UIR system. Finally, Section 4 discusses conclusions and future work.

2. User Intent Recognition Framework

In this section, we present the methodology used to design the UIR system. The architecture of the UIR system is illustrated in Figure 1. Our new contribution is a novel feature selection method based on MOO, as illustrated in Figure 1 in the double-lined box (Section 2.4). In this box, the application of four MOBBO methods for gait mode classification is new, and a novel MOO-based feature selection method called GMOFS is new. The remaining parts of the UIR system are implemented based on the existing literature. The role of each subsystem is explained in more detail in the following subsections.

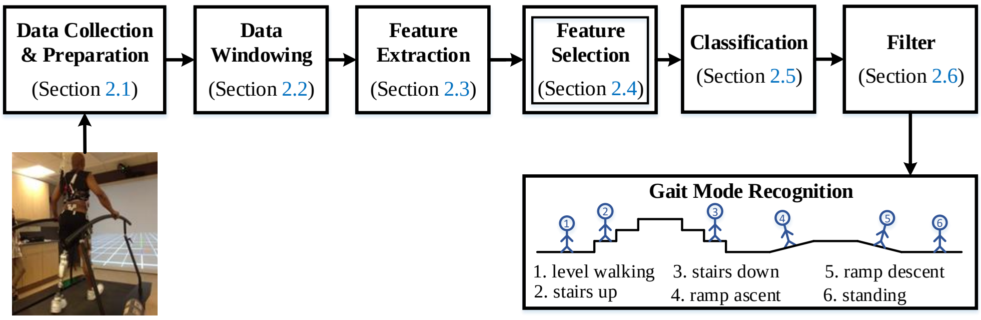


Figure 1. Architecture of user intent recognition system. The double-lined box indicates that an evolutionary algorithm is used for optimization.

2.1. Data Collection and Preparation

Data collection can significantly affect the accuracy of UIR. Input signals must be informative enough to accurately discriminate between various human gait modes. Four types of input data are commonly used in the literature for the gait mode recognition of transfemoral amputees [12,13]. These four types include signals that reflect: (1) the state of the prosthesis, (2) the state of the residual limb, (3) the user-prosthesis interaction, and (4) the prosthesis-environment interaction.

In this paper, we collect vertical hip position and thigh angle to indicate the state of the residual limb, and hip moment to indicate the user-prosthesis interaction. These signals are like an implicit communication link between the user and the prosthesis and can be used to infer user intent. Based on Nyquist's sampling theorem, measurement signals are sampled at 100 Hz, a frequency that is significantly higher than that of human walking [33]. Then the signals are filtered to eliminate noise and to handle missing measurement data points [34]. Missing data points are recovered with basic interpolation methods.

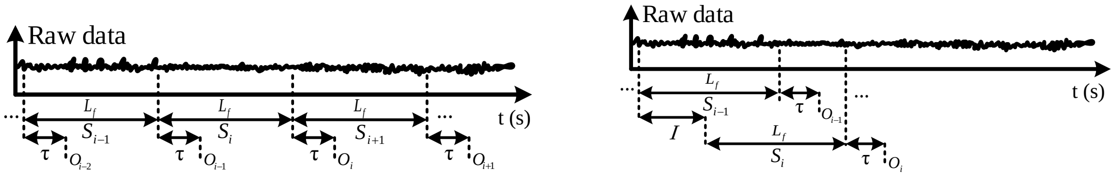
2.2. Data Windowing

To effectively classify human gait modes, we extract appropriate features from a frame (window) of measurement signals. Frame length L_f defines the number of samples in a frame. A short frame fails to provide a rich data set and may lead to significant classification bias and variance. On the other hand, a long frame is a computational burden for real-time implementation. In this paper, L_f is chosen to trade off feature richness and computational load.

We apply two different methods for data windowing: disjoint windowing and overlapped windowing [35]. Figure 2 illustrates the two windowing approaches. In disjoint windowing, the class outcome O_i corresponding to frame S_i is output every L_f ms. τ is the time required for feature extraction, classification, commanding the appropriate low-level controller, and prosthesis response time. In overlapped windowing, we use a sliding frame with length L_f and increment I, and the class outcome is output every I ms. Disjoint windowing is a special case of overlapped windowing when $I = L_f$. To achieve real-time operation, the parameters of the windowing approaches should satisfy

$$\begin{aligned} &\tau \le L_f && \text{disjoint windowing} \\ &\tau \le I \le L_f && \text{overlapped windowing} \end{aligned} \tag{1}$$

In this paper, we apply disjoint and overlapped windowing with various frame and increment lengths. We consider two important characteristics to determine L_f [35]: (1) the minimum interval between two distinct muscle contractions is 200 ms [36], and (2) the delay between user intent and the resultant prosthesis motion should be no more than 300 ms [37,38]. The first property implies that a 200 ms frame of data should have the potential to provide rich features for gait mode classification. The second property, which is known as the real-time constraint, ensures that the amputee will experience the prosthesis as responsive to his or her intent. The real-time constraint requires $\tau \le L_f \le 300$ ms for disjoint windowing, and $I \le 300$ ms for overlapped windowing. Therefore, we use overlapped windowing when the frame length is larger than 300 ms, noting that a larger frame will require a higher computational load.



(a) Disjoint windowing **(b)** Overlapped windowing

Figure 2. Data windowing. S_i represents the i-th data segment, L_f is the frame length, τ is the required processing time, I is the increment length for overlapped windowing, and O_i is the detected gait mode corresponding to frame S_i.

2.3. Feature Extraction

Various features can be extracted from a frame of measurement data and used for classification. Features should be rich enough to discriminate between various gait modes. In addition, feature extraction needs to be computationally fast for real-time implementation. In general, both time-domain (TD) and frequency-domain (FD) features are frequently used for classification [35,39,40]. We compare TD and FD features in this paper, and select the optimal subset of features for UIR.

TD features are computationally fast, and include information about the data waveform and frequency. We extract the following TD features from each frame of data: slope sign change (SSC), zero crossing (ZC), waveform length (WL), variance (VAR), mean absolute value (MAV), modified MAV, root mean square (RMS), Willison amplitude (WAMP), skewness (SK), kurtosis (KU), and correlation coefficient (COR) and angle (ANG) between two frames of data from different measurement signals.

In addition, multiple FD features have been extracted. FD features are computationally slower than TD features, but include information about the frame's frequency content. We use periodograms to measure the power spectrum density (PSD) of a frame, and calculate the following FD features: mean frequency (MNF), median frequency (MDF), maximum frequency (MAXF), and fourth-order auto-regressive coefficients (AR4).

Previous research has shown the applicability of these TD and FD features for prosthetic limb pattern recognition [39–41]. Therefore, we are motivated to investigate the performance of these features for gait mode recognition. The mathematical definitions of these TD and FD features are given in [39].

After extracting TD and FD features from a frame of measurement data, the features are concatenated and labeled to create a single training pattern. For instance, extraction of VAR, MAV + RMS, and AR4 features from a frame of three measurement signals (e.g., vertical hip position, thigh angle, and thigh moment) would produce a training vector with 3, 6, and 12 elements, respectively. We perform the above procedure for all features and all frames of measurement data to create the training data set. The training set is then normalized to equalize the relative magnitude of each feature.

2.4. Feature Selection

The objective of feature selection is to find a subset of the features that were obtained with the feature extraction method. The feature selection method attempts to find a parsimonious feature subset that results in accurate classification. However, subset size and classification accuracy are conflicting objectives. A small feature subset will probably result in high classification error, whereas a large feature subset will probably result in lower classification error. Therefore, feature selection can be viewed as an MOO problem. In MOO problems, no single solution can simultaneously optimize all objectives. The solutions comprise a set of possible alternative solutions known as the optimal Pareto set [22].

We seek the most informative but parsimonious subset of features for gait mode classification. Note that exhaustive search is not practical in cases with a high-dimensional set of features. A set of n features has 2^n different subsets. Many heuristic search strategies, such as sequential forward selection, sequential backward elimination, and evolutionary search, have been suggested for this type of combinatorial problem [42]. EAs have been demonstrated as an efficient search strategy for feature selection [43]. In this section, we propose a search strategy based on BBO, in addition to a new gradient-based algorithm, to solve the MOO problem. We then use two systematic approaches to compare the performance of the different search strategies.

2.4.1. Biogeography-Based Multi-Objective Optimization

We propose the application of MOBBO for feature selection of the UIR system. The dimension of the optimization problem is equal to the number of available features (independent variables). Each feature is represented by a binary value where 1 indicates that the feature is used for classification, and 0 indicates otherwise. Therefore, each individual in the MOO algorithm is a binary sequence with length equal to the problem dimension. We evaluate the following two objective functions for all individuals in the population.

$$\begin{aligned} f_1^i &= \text{number of selected features in the } i\text{-th individual} \\ f_2^i &= \text{average prediction error using } c\text{-fold cross validation} \end{aligned} \tag{2}$$

where $i = 1, \cdots, N$, and N is the population size. We combine BBO with four MOO algorithms [22] to obtain VEBBO, NSBBO, NPBBO, and SPBBO. We apply these MOBBO variants to find the optimal Pareto set for the optimization problem. We investigate the performance of each method in a later section.

Although it is possible to use any classification algorithm, we use LDA to compute f_2^i. LDA, unlike other learning algorithms such as MLP, does not require time-consuming iterations to build a model. This point is important because EAs require many objective function evaluations to find the solution. It is possible to use either classification accuracy or error as the quality measure for the second objective. We use average classification error of c-fold cross validation (CV). In c-fold CV, we randomly divide the training set into c distinct folds. Then we repeat training c times; each time the model is trained using $c-1$ folds and is tested with the remaining fold. The average of the c classification errors is used as the quality measure.

2.4.2. Gradient-Based Multi-Objective Feature Selection

Although MOBBO and other gradient-free MOO methods have the potential to find the globally optimal solution [23,24], they are computationally expensive due to the need for many iterations of the classifier training process (multiple individuals in the population, and multiple generations). To reduce computational complexity, we propose a novel algorithm called GMOFS for feature selection. In GMOFS, feature selection and data classification are performed simultaneously.

GMOFS incorporates a regularization penalty term to the optimization problem of its learning algorithm. The penalty term, which is handled by a Lagrange multiplier, directs the trained model toward a parsimonious as well as accurate model. We use an MLP network as the classifier, and include an elastic net to penalize the size of the selected feature subset. The first step of GMOFS is to train a constrained MLP network with the cost function

$$J = \frac{1}{2} \sum_{l=1}^{m} \sum_{j=1}^{K} \left(t_j^{(l)} - o_j^{(l)} \right)^2 + \lambda \sum_{i=1}^{n} \left(\alpha \beta_i^2 + (1-\alpha)|\beta_i| \right) \tag{3}$$

where β_i is the multiplier of the i-th input feature before input to the MLP network; $t_j^{(l)}$ and $o_j^{(l)}$ are the target and actual value, respectively, of the j-th output neuron associated with the l-th training pattern; K is the number of output neurons (classes); m is the number of training patterns; and n is the number of input features. We use an MLP network with one hidden layer and p hidden neurons (including the bias node). v_{ih} denotes the weight that connects the i-th input neuron to the h-th hidden neuron, and w_{hj} denotes the weight that connects the h-th hidden neuron to the j-th output neuron. The first term of the cost function penalizes classification error while the second term, which is the elastic net, penalizes the number of selected features. The elastic net is a convex combination of ridge regression ($\alpha = 1$) and least absolute shrinkage and selection operator ($\alpha = 0$). $\lambda \geq 0$ is a complexity parameter that controls the shrinkage of the input features. Large λ leads to a shrinkage of β_i toward zero, which implies that the input feature corresponding to β_i is not significant. However, as shrinkage increases, classification error tends to increase. Therefore λ provides a trade-off between classification error and the number of selected features. In summary, the construction of the MLP network with the elastic net is formulated as the following optimization problem:

$$\min_{\beta,w,v} J \;\text{ subject to }\; \begin{cases} 0 \le \beta_i \le 1 \\ |w_{hj}| \le a \text{ and } |v_{ih}| \le b \end{cases} \tag{4}$$

for all i, j, h, where $\beta_i = 0$ or 1 implies that the associated feature is the least or most significant input variable, respectively. However, due to the direct relationship between β_i and neuron weights v_{ih} and w_{hj}, we cannot conclude that an input feature with small β_i and large neuron weights is insignificant. To avoid optimization solutions with large weights, neuron weights are constrained. Backpropagation is used to update β_i, v_{ih}, and w_{hj}. The derivative of J with respect to output weights w_{hj}, hidden weights v_{ih}, and input weights β_i is obtained by the chain rule as

$$\begin{aligned} \frac{\partial J}{\partial w_{hj}} &= \sum_{l=1}^{m} \delta_j^{(l)} y_h^{(l)} \\ \frac{\partial J}{\partial v_{ih}} &= \sum_{l=1}^{m} \left[\sum_{k \in D_2(h)} \left[\delta_k^{(l)}\, w_{hk}\right] y_h^{(l)} (1 - y_h^{(l)})\; \beta_i x_i^{(l)} \right] \\ \frac{\partial J}{\partial \beta_i} &= \sum_{l=1}^{m} \left\{ \sum_{s \in D_1(i)} \left[\sum_{k \in D_2(h)} \left[\delta_k^{(l)}\, w_{sk}\right] y_s^{(l)} (1 - y_s^{(l)}) v_{is} \right] x_i^{(l)} \right\} + \lambda \left[2\alpha\beta_i + (1-\alpha) \frac{\beta_i}{|\beta_i|} \right] \end{aligned} \tag{5}$$

where $D_1(i)$ is the set of middle layer neurons whose inputs come from the i-th input neuron, $D_2(h)$ is the set of output neurons whose inputs come from the h-th middle layer neuron, and $\delta_j^{(l)} = -(o_j^{(l)} - t_j^{(l)})(1 - t_j^{(l)})t_j^{(l)}$. We use the derivatives in Eq. 5 and constraints in Eq. 4 along with the trust region reflective algorithm to train the constrained MLP network.

Once MLP training phase is completed, input weights β_i are sorted in descending order. The input variable with the largest β_i is the most significant feature. The second step of GMOFS is to select the most r significant features, which are associated with the r largest input weights β_i, and which satisfy

$$\begin{gathered} \frac{\sum_{i=1}^{r} \beta_i}{\sum_{j=1}^{n} \beta_j} \ge 95\% \\ \beta_1 \ge \beta_2 \ge \cdots \ge \beta_n \end{gathered} \tag{6}$$

We then repeat the first two steps of GMOFS for different λ in the range $[\lambda_l, \lambda_u]$ with a predefined increment $\triangle\lambda$. The selected subsets associated with each λ comprise a population. The population size depends on $\triangle\lambda$. To assess the performance of the selected feature subset, we train a classifier with each selected subset and find classification error. In this population, the subset associated with $\lambda \to \infty$ has minimum size and maximum classification error, whereas the subset with $\lambda = 0$ has maximum size and probably has the lowest classification error. Thus, the size of the selected subset and the classification error, defined in Eq. 2, are two conflicting objectives. To find the GMOFS Pareto front, we first obtain the Pareto set as

$$P_s = \left\{ x^* : \left[\nexists x : f_i(x) \le f_i(x^*) \text{ for all } i \in [1,2], \text{ and } f_j(x) < f_j(x^*) \text{ for some } j \in [1,2] \right] \right\} \tag{7}$$

x^* denotes the set of non-dominated solutions in the population and $f_i(x)$ is the i-th objective function. The Pareto front P_f is obtained from all function vectors $f(x)$ that correspond to the Pareto set:

$$P_f = \{ f(x^*) : x^* \in P_s \} \tag{8}$$

Note that all Pareto points are equally preferable apart from subjective prioritization. The outline of GMOFS is given in Algorithm 1.

Algorithm 1 The outline of gradient-based multi-objective feature selection (GMOFS), where x_i is the i-th feature in the training set X, and Y is the corresponding set of output classes.

```
Initialization: $\lambda = \lambda_l \leq \lambda_u$, Population = $\emptyset$, $k = 1$
While $\lambda \leq \lambda_u$
    Step 1:
    Use the training data $\{X, Y\}$ to train the constrained MLP network in Eq. 3 by solving Eq. 4
    Step 2:
    Sort the input weights $\{\beta_i\}$ in descending order
    Use Eq. 6 to select subset $S_k \subset X$ where size($S_k$) $\leq$ size($X$)
    Step 3:
    Population $\leftarrow$ Population $+ S_k$
    $k \leftarrow k + 1$
Next $\lambda \leftarrow \lambda + \triangle\lambda$
Step 4:
For each subset $S_k$ in Population
    Use cross-validation to train and test a classifier with dataset $\{S_k, Y\}$
    Calculate objective functions $f_1^k$ and $f_2^k$ using Eq. 2
Next subset $S_k$
Step 5:
Find the Pareto set using Eq. 7
```

2.4.3. Evaluation of Multi-Objective Optimization Pareto Fronts

We will compare the Pareto fronts obtained by each MOO algorithm using normalized hypervolume and relative coverage. These methods are popular for evaluating the quality of a Pareto front. The Pareto front normalized hypervolume is computed as follows:

$$\text{Normalized Hypervolume} = \frac{\sum_{j=1}^{N_p} \prod_{i=1}^{M} f_{ji}}{N_p} \tag{9}$$

where M is the number of objectives, f_{ji} is the value of the i-th objective function of the j-th Pareto point, and N_p is the number of Pareto points.

Another way to compare Pareto sets is to compute the coverage of one Pareto set relative to a second Pareto set. This metric is determined by the number of solutions in the first Pareto set that are weakly dominated by at least one solution in the second Pareto set [22]. A smaller number for normalized hypervolume and relative coverage indicates better performance.

2.5. Classification

Accurate classification of gait patterns is the ultimate goal of the UIR system. For this purpose, we assess various well-known linear and non-linear classification techniques, including LDA, QDA, SVM, DT, and MLP classifiers.

LDA and QDA classifiers do not require time-consuming iterations for training. In fact, the parameters of these classifiers are directly obtained from the training data. Although these classifiers are fast in terms of training, they are not as flexible as nonlinear classifiers such as SVM, DT, and MLP. These classifiers solve an optimization problem that minimize the classification error. In most cases, it is difficult to find optimization solutions in closed form, so we use either iterative optimization algorithms such as steepest descent, or EAs.

We use one-against-one approach to implement multi-class SVM, and we also evaluate the performance of different kernels, such as linear and RBF. We tune the parameters of the SVM kernels to achieve the best classification performance. To increase the accuracy of the MLP network, we perform a grid search of the number of hidden nodes p from the set $\{3, 4, 5, 6, 8, 10, 15, 20\}$, and we measure the mean classification error using five-fold CV. Then we choose p to obtain a trade-off between classification accuracy and classifier complexity. An MLP with small p may not result in the desired accuracy, but an MLP with large p may tend to memorize the noise in the training set and lead to overfitting and poor generalization. In addition, we use Wilcoxon signed-rank tests to statistically compare the classification methods.

2.6. Filter

To enhance the prediction performance of the UIR system, we incorporate a majority voting filter (MVF). MVF alleviates transient jumps between classifier output classes and leads to smooth transitions from one classified gait mode to another [13]. We implement the MVF using $2q + 1$ classified modes [41]: the current, q previous, and q subsequent values. The MVF output is the most frequently classified mode among those $2q + 1$ values.

Since MVF uses q future classified gait modes to calculate the current classified mode, an inappropriate value for q may violate the real-time constraint. The constraint requires that the classified mode is output with a time delay less than 300 ms (see Section 2.2). The constraint requires

$$\begin{aligned} q \times L_f &\le 300 \text{ ms} \qquad \text{Disjoint windowing} \\ q \times I \ \ &\le 300 \text{ ms} \qquad \text{Overlapped windowing} \end{aligned} \tag{10}$$

An MVF with very small q may not significantly improve classification performance, whereas an MVF with very large q may cause misclassification because of time delay. In this paper, we will choose a trade-off value for q, and will investigate the effect of the MVF on classification performance.

3. Results and Discussion

This section evaluates the performance of the UIR system and its subsystems as discussed in Section 2.

3.1. Experimental Setup and Data Collection

To design and evaluate the performance of the UIR system, we collect data from three able-bodied subjects (AB01, AB02, and AB03) and three transfemoral amputee subjects (AM01, AM02, and AM03). All the experiments were approved by the Department of Veterans Affairs Institutional Review Board. The above-knee amputees wore an Ottobock prosthesis on the right leg. As discussed in Section 2.1, we measure vertical hip position, thigh angle, and thigh moment for UIR. These signals are collected for able-bodied subjects during four different activity modes: (1) standing (ST), (2) normal walking (NW) at user-preferred speed (PS), (3) slow walking (SW), and (4) fast walking (FW). Due to physical limitations, we collect data during only three activity modes for the amputee subjects: ST, NW, and SW. Table 1 shows the physical characteristics of the subjects.

The data were collected at the Motion Study Laboratory of the Cleveland Department of Veterans Affairs Medical Center with 47 reflective markers on each subject’s body. Subjects were asked to walk on a treadmill with built-in force sensors. A 16-camera Vicon system recorded kinematic data at 100 Hz. Ground reaction force along three axes were collected from the force sensors at 1000 Hz. Data were filtered with a second-order low-pass filter with a cutoff frequency of 6 Hz. Filtered data were input to inverse dynamics software to calculate the joint angles and torques. Detailed methods and sample results can be found in [34]. The experimental setup is illustrated in Figure 3. Note that the lower-limb amputee demographic at the Veterans Affairs Medical Center, where data collection was performed, is dominated by males. Over 98% of veterans who underwent amputation in 2011 were male [44].

Table 1. Physical characteristics of the six human test subjects. AB and AM represent able-bodied and amputee subject, respectively.

	Gender	Age (years)	Weight (kg)	Height (cm)	Walking Speed (m/s) SW	PS	FW
AB01	Male	37	79.5	188	0.98	1.30	1.63
AB02	Male	20	73.9	172	0.86	1.15	1.44
AB03	Male	28	80.9	179	0.75	1.00	1.25
AM01	Male	32	79.1	174	0.60	1.00	–
AM02	Male	64	99.2	177	0.56	0.94	–
AM03	Male	35	81.7	176	0.60	0.90	–

Our future work will need to include more subjects and wider demographics (for example, ages and genders).

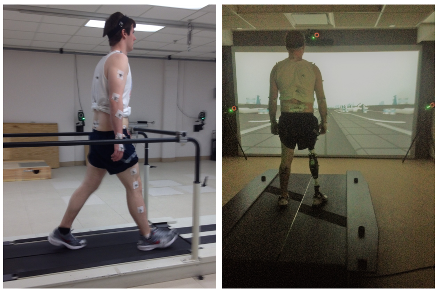


Figure 3. Experimental setup for data collection. The left figure shows an able-bodied subject and the right figure shows an amputee subject with an Ottobock prosthesis on the right leg.

Note that in a real-world, non-laboratory settings, we would measure the required input signals directly rather than with cameras. For example, we could use piezo-electric sensors or multi-axis load cells for force sensing, and optical encoders or inertial measurement units for position and angle sensing [12].

The able-bodied subjects were asked to perform four sequences of walking trials, each lasting approximately 60 seconds. Each sequence consists of four different gait modes (ST, SW, NW, and FW) and each mode was maintained for several seconds. Figure 4 illustrates a sample walking trial for AB01. The amputee subjects performed six sequences of three different walking modes, each lasting approximately 30 seconds.

In summary, we note a few important points. Firstly, in this paper, the type of walking activities used for recognition is not our main focus, but rather the assessment of the proposed methodology to eliminate irrelevant/redundant features for UIR is our main goal. Secondly, in human activity mode recognition applications, an entire stride is typically used for non-real-time classification [45]. However, in UIR, we use a small window of measurement signals, mostly within a few milliseconds, to identify user's intent for real-time prosthesis control.

3.2. Effect of Frame Length on Classification Performance

The objective of this section is to choose the appropriate data windowing method and frame length. We investigate the influence of disjoint and overlapped windowing with different frame lengths on the classification accuracy of the UIR system. We use disjoint windowing with frame lengths $L_f = \{100, 150, 200, 250\}$ ms, and overlapped windowing with frame lengths $L_f = \{200, 250, 300\}$ ms and increments $I = \{50, 150, 200\}$ ms. We extract TD and FD features from the data frames generated by the three measurement signals collected from able-bodied subjects. Figure 5 shows the mean classification performance of LDA with different frame lengths. Note that the size of the dataset depends on the parameters of the windowing methods. For instance, a 10-second walking sequence with disjoint windows of length 100 ms provides 100 frames and consequently 100 training patterns.

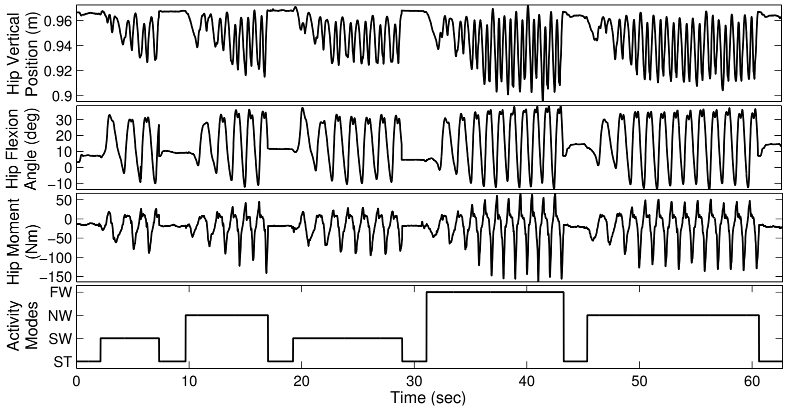


Figure 4. Sample walking trial with four different gait modes for able-bodied subject AB01

In this experiment, LDA is separately trained and tested using 10-fold CV for each subject with only one feature type for each of the measurements: WL, VAR, MAV, RMS, WAMP, ANG, and AR4. These features are used because they are considered the most representative TD and FD features. Figure 5 illustrates the mean classification accuracy of LDA for the two able-bodied subjects using 10-fold CV. A single value on the horizontal axis of the figure indicates the frame length of disjoint windowing. A pair of values indicates the frame length and increment length of overlapped windowing; for instance, 200−50 denotes $L_f = 200$ ms and $I = 50$ ms.

Figure 5 shows that the classification accuracy typically improves as the frame length increases. A larger frame is more likely to include richer information, and consequently lower bias and variance in classification performance. For instance, the increase in accuracy with WL is approximately 19% when the frame length increases from 100 ms to 200 ms. The increases are 14.3% and 16% when using the VAR and AR4 features, respectively. However, the accuracy does not vary significantly with frame length for the remaining features in Figure 5. The figure illustrates that all representative TD features except ANG provide better classification performance than AR4, which is the representative FD feature.

In this experiment, very small frame length is not used because it would result in poor prediction accuracy. Conversely, large frame length is not used because it would result in a violation of the real-time constraint. To find the best frame length, we statistically compare performance using

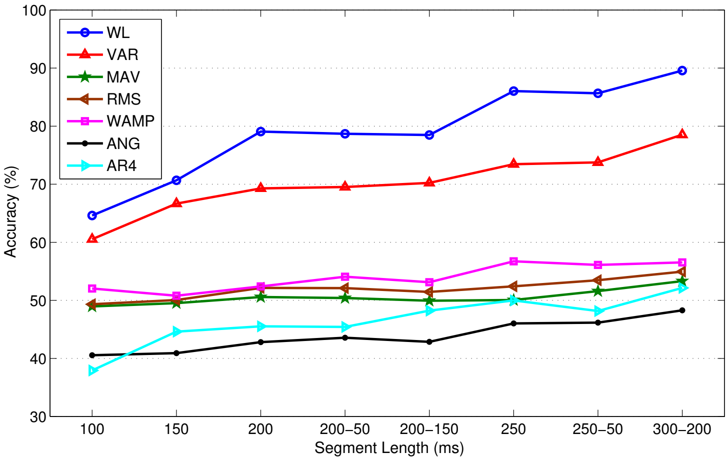


Figure 5. Mean LDA performance for the able-bodied subjects for different data windowing methods and frame lengths. On the horizontal axis, a single value indicates the frame length of disjoint windowing, and a pair of values indicates the frame length and increment length of overlapped windowing. For instance, 200−50 indicates $L_f = 200$ ms and $I = 50$ ms for overlapped windowing.

Wilcoxon signed-rank tests. For this purpose, LDA is trained using only one TD or FD feature at a time. The null hypothesis of the test is that the differences between mean classification accuracy corresponding to two different frame lengths are from a distribution with zero mean at the specified level of significance. If the null hypothesis cannot be rejected, then we conclude that the two compared frame lengths are not statistically significantly different, as indicated by an $\approx$ sign and a T (tie). If we can reject the null hypothesis, then the two frame lengths are statistically significantly different, and this is indicated by a $+$ sign. The better frame length is the one with better mean classification accuracy and is shown by B (better) while the worse one is shown by W (worse). Table 2 provides the results of the statistical tests at a 10% significance level.

Table 2 shows that frames with length larger than 200 ms perform better than frames with length 150 or 100 ms. Table 2 shows that the two overlapped frame windows with $L_f = 250$ ms, $I = 50$ ms and $L_f = 300$ ms, $I = 200$ ms tie for similar performance, and perform better than the other frame lengths.

Taking into account the real-time constraint, the length of the MVF filter, and processing time, we choose overlapped windowing with $L_f = 250$ ms, $I = 50$ ms throughout the remainder of the paper as the best trade-off, except where specifically mentioned otherwise.

Table 2. Comparison of classification performance for different frame lengths (row values versus column values) using Wilcoxon signed-rank tests at a 10% significance level. $\approx$ indicates that the two compared frame lengths tie (T) with similar performance and are not statistically significantly different. $+$ indicates that the two frame lengths are statistically significantly different, and B or W indicates that the row frame length performs better or worse than the column frame length, respectively.

Frame Length (ms)		150	200	200-50	200-150	250	250-50	300-200
100	vs.	W (+)	W (+)	W (+)	W (+)	W (+)	W (+)	W (+)
150	vs.	–	W (+)	W (+)	W (+)	W (+)	W (+)	W (+)
200	vs.	*	–	W (+)	T ($\approx$)	W (+)	W (+)	W (+)
200-50	vs.	*	*	–	B (+)	W (+)	W (+)	W (+)
200-150	vs.	*	*	*	–	W (+)	W (+)	W (+)
250	vs.	*	*	*	*	–	W (+)	W (+)
250-50	vs.	*	*	*	*	*	–	T ($\approx$)

We use principal component analysis (PCA) [46] and Fisher linear discriminant analysis (FLDA) [47] to visualize the training set. A training pattern is a vector of all TD and FD features extracted from a frame of raw data. We performed 2D dimension reduction for three different frame lengths. Figure 6 illustrates the 2D scatter plot for able-bodied subject AB01. To save space, we do not provide the same figures for other subjects because we obtained similar results for those subjects.

Figure 6 shows that FLDA provides better visualization than PCA in terms of gait mode separability. We verify that longer frame length leads to better gait mode separation, and eventually better classification performance. Most importantly, Figure 6 verifies the effectiveness of the TD and FD features.

3.3. Multi-Objective Feature Selection

We perform feature selection in two steps. In the first step we exclude non-informative TD and FD features, and in the second step we use MOO to further refine the selected feature set. We implement five different MOO methods, including our newly proposed method, to find the optimal set of features. The complete set of features includes the following TD and FD features: SSC (F_1), ZC (F_2), WL (F_3), VAR (F_4), MAV (F_5), MAV1 (F_6), MAV2 (F_7), RMS (F_8), WAMP (F_9), SK (F_{10}), KU (F_{11}), FMD (F_{12}), FMN (F_{13}), FMAX (F_{14}), COR (F_{15}), ANG (F_{16}), and AR4 (F_{17}). We perform preliminary feature selection using two able-bodied subjects. We use MOO for final feature selection using only one able-bodied subject to reduce computational effort. Later we will investigate the performance of the selected features with other able-bodied subjects and with the amputee subjects.

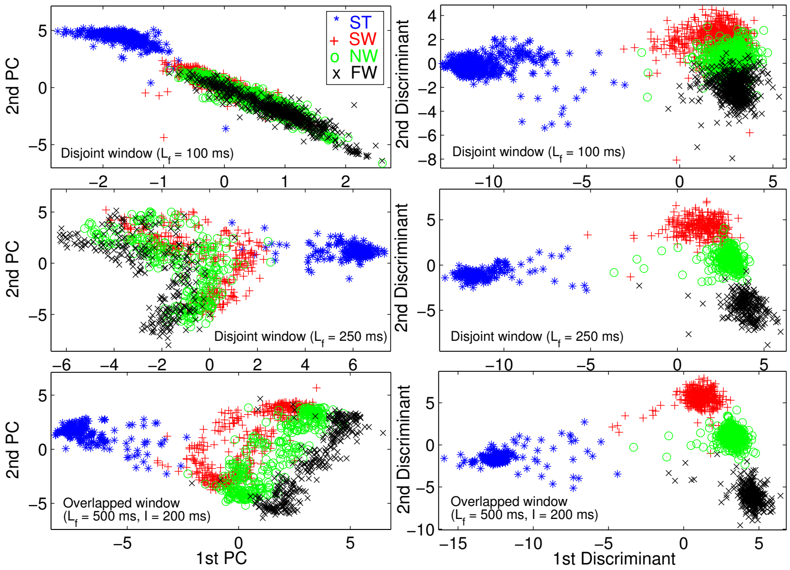


Figure 6. Two-dimensional scatter plot for visualization using PCA (left column) and FLDA (right column) for able-bodied subject AB01

In the first step, we train LDA for each of two able-bodied subjects, and separately for each individual feature type listed in the previous paragraph, using 10-fold CV for each training procedure. The mean classification accuracy over the two subjects and the ten folds are used to assess the importance of each feature type. Since we need to train the classifier several times in this step, we choose to use LDA due to its fast training time.

Figure 7 shows the mean classification accuracy and processing ratio for each feature type. Processing ratio indicates the relative computational complexity to compute each feature type. Figure 7 indicates that TD features require less computational effort than FD features. For instance, F_{12}, F_{13}, and F_{14} require high computational effort compared to other features. We exclude F_{12}, F_{13}, and F_{14} from the candidate feature set due to their relatively high computational expense and poor classification accuracy. In addition, F_6 and F_7, two variants of MAV, are excluded due to their poor classification accuracy and because they provide information that is similar to MAV. Therefore, we exclude a total of five weak feature types, and pre-select the remaining 12 feature types. This results in a training vector with 11×3+4×3=45 elements. Note that the AR4 feature type includes four components and thus contributes a total of 12 elements from the three measurement signals. Finally, vertical hip position does not cross zero (see Figure 4), thus the ZC feature of this signal is excluded. In summary, we have a training data set with 44 features.

Now we are ready to proceed to the feature selection step. In this step we use VEBBO, NSBBO, NPBBO, SPBBO, and GMOFS to select an optimal subset from the 44 pre-selected features. To reduce the computational expense, we use only the AB01 training data set in this step. We then verify that the selected subset results in a satisfactory UIR system when trained for other subjects. Table 3 shows the tuning parameters used in this paper. To tune the parameters, we performed a sensitivity analysis of MOO performance to each parameter, one at a time, to find a local optimum of MOO performance with respect to each parameter. For instance, GMOFS is implemented with different elastic net parameter values $\alpha = \{0, 0.5, 1\}$, and we found that the Pareto front with $\alpha = 0$ dominates Pareto fronts that are found with other values of α. For training the neural network in GMOFS, we used the MATLAB function fmincon from the Optimization Toolbox to implement a trust region reflective algorithm. We mostly used default values for the fmincon parameters, but we found that the performance of GMOFS is not very sensitive to these parameters.

We run each multi-objective method for 10 independent trials, and the best Pareto front of each method is selected for MOO comparison. Results show that the GMOFS Pareto front statistically

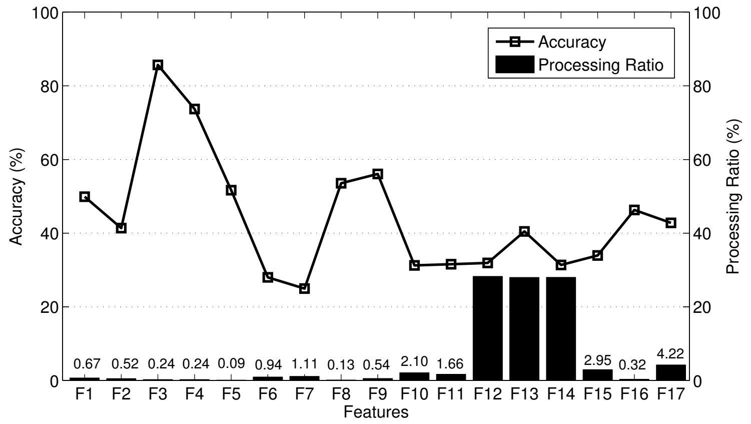


Figure 7. Mean classification accuracy over two able-bodied subjects, and processing ratio of 17 feature types trained by LDA using 10-fold cross validation

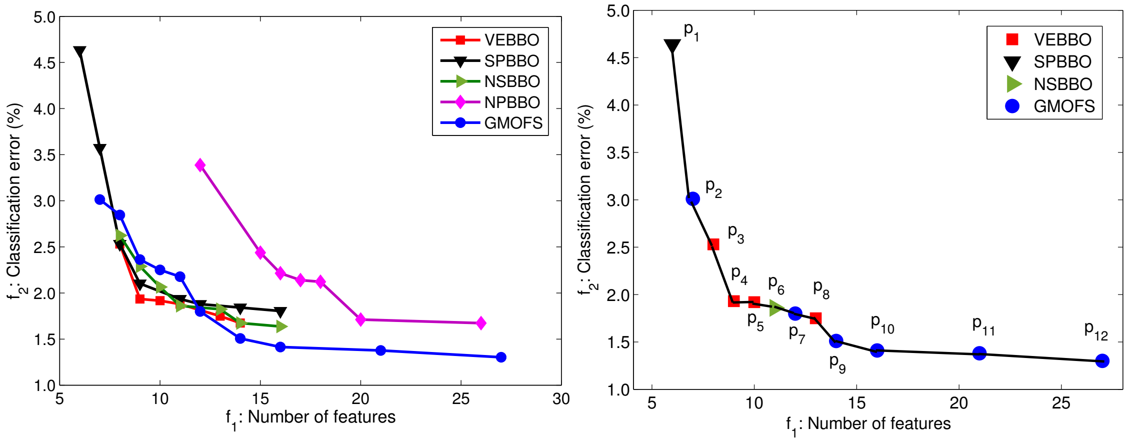
Table 3. Tuning parameters for multi-objective feature selection

	Symbol	Value
MOBBO		
Mutation rate	μ	0.04
Number of elites	E	2
Population size	N	100
Number of generations	Gen	1000
Problem dimension	d	44
Migration model	m_{flag}	sinusoidal
GMOFS		
Number of hidden nodes	p	5
Elastic net parameter	α	0
Bound for shrinkage parameter	$[\lambda_l, \lambda_u]$	[0, 150]
Bound for neuron weights	a, b	5
Increment of shrinkage parameter	$\triangle\lambda$	1 if $0 \leq \lambda \leq 30$; and 10 if $30 < \lambda \leq 150$
Trust region reflective		
Maximum allowable iterations	MaxIter	100
Termination tolerance on the independent variable	TolX	1E–3
Termination tolerance on the cost function	TolFun	1E–3
Typical values for the independent variable	TypicalX	0.1
Finite difference method	FinDiffType	central

significantly dominates all four MOBBO Pareto fronts. However, we are aware of the fact that MLP, which is used in GMOFS, generally outperforms LDA, which is used in the MOBBO variants, for complicated nonlinear problems.

To have a more fair comparison, we apply SVM with linear kernels to all of the optimal feature subsets found by the MOO methods. Figure 8(a) illustrates the Pareto fronts obtained by the five MOOs with SVM with linear kernels. Figure 8(a) shows that the Pareto fronts of VEBBO, SPBBO, NSBBO, and GMOFS are close, and clearly dominate the NPBBO Pareto points. Figure 8(b) indicates the combined Pareto front obtained from all of the non-dominated points in Figure 8(a). GMOFS provides the maximum contribution to the combined Pareto front, while NPBBO does not contribute any Pareto points. All of the points in Figure 8(b) are labeled for easy referencing.

To systematically compare the Pareto fronts in Figure 8(a), we use relative coverage and normalized hypervolume as discussed in Section 2.4.3. Tables 4 and 5 provide the comparison results using these two approaches. In Table 4, an entry in column i and row j ($i \neq j$) indicates the percentage of Pareto points of the method of column i that is dominated by at least one Pareto point of the method of row j. We see that on average, only 7.2% of the Pareto points of VEBBO are weakly dominated by



(a)

(b)

Figure 8. (a) Pareto fronts obtained from MOO methods with an SVM classifier with linear kernels using AB01 training data; (b) combined Pareto front obtained from non-dominated Pareto points in (a).

at least one Pareto point from the other four MOO methods. Therefore, VEBBO ranks first in terms of relative coverage. GMOFS ranks second and performs better than SPBBO, NSBBO, and NPBBO. In addition, Table 5 shows that VEBBO and GMOFS rank first and second in terms of normalized hypervolume, respectively. GMOFS ranks first in terms of the number of Pareto points. These results verify the competitive performance of GMOFS compared to the other four MOO methods.

Most importantly in terms of the advantage of GMOFS, it requires the execution of only 43 classifier training procedures (due to the number of λ increments), while each of the other four EA-based MOO methods require 100,000 training procedures (due to the combination of population size and generation limit).

Table 4. Comparison of Pareto fronts using relative coverage (RC). Only 7.2% and 30% of the VEBBO and GMOFS points, respectively, are dominated by other Pareto points; so VEBBO and GMOFS rank first and second, respectively, in terms of RC.

	VEBBO	SPBBO	NSBBO	NPBBO	GMOFS
VEBBO	–	62.5	75.0	85.7	40.0
SPBBO	0.0	–	25.0	71.4	40.0
NSBBO	14.3	50.0	–	100.0	40.0
NPBBO	0.0	0.0	0.0	–	0.0
GMOFS	14.3	50.0	50.0	100.0	–
Mean RC (%)	**7.2**	40.4	37.5	89.3	30.0

Table 5. Comparison of Pareto fronts using normalized hypervolume. N_p is the number of Pareto points obtained by each MOO method. VEBBO and GMOFS rank first and second, respectively, in terms of normalized hypervolume, and GMOFS ranks first in terms of the number of points.

	N_p	Normalized Hypervolume
VEBBO	7	**0.5026**
SPBBO	8	0.5814
NSBBO	8	0.5676
NPBBO	7	0.8013
GMOFS	**10**	0.5332

The benefit of presenting the data of Figure 8(b) is that it allows us to find the best subset of features for an accurate and parsimonious classifier. Among the 12 Pareto points, we choose p_9 as a potential candidate solution. We could pick any other solution from the Pareto front depending on the priority of the problem objectives, but p_9 provides a good trade-off between classification error and number of features. Therefore, in Section 3.5 we will investigate classification performance with candidate solution p_9 for all human subject data AB01, AB02, AB03, AM01, AM02, and AM03. However, first we will find the best classifier in the following section.

3.4. Comparison Results of Classification Algorithms

In this section, we use p_1 through p_{12} to statistically compare the performance of different classifiers for subject AB01. The objective is to find the best classifier for locomotion mode detection among LDA, QDA, SVM with linear kernels (SVM-linear), SVM with RBF kernels (SVM-RBF), MLP, and DT. The tuned parameter value of RBF kernel is $\sigma = 1$. Table 6 shows mean classification accuracy and standard deviation of each classifier trained with the features from each Pareto point using 10-fold CV.

Table 7 presents pair-wise statistical comparisons using Wilcoxon signed-rank tests at a 5% significance level. If a pair-wise p-value is less than 0.05, the mean performances of the two classifiers are statistically significantly different, and the classifier with larger mean prediction accuracy performs better than the other one. A pair-wise p-value greater than 0.05 indicates no significant difference between the performance of the two classifiers. Table 7 shows that the classification performance of MLP and SVM-RBF are statistically equal, and are significantly better than the other methods. SVM-linear is statistically better than LDA, QDA, and DT. QDA performs better than LDA and similarly to DT. In summary, MLP and SVM-RBF are the best, SVM-linear is the second best, QDA and DT are the third best, and LDA is the worst for locomotion mode detection.

Table 6. Mean classification accuracy (ACC) and standard deviation (STD) for AB01 of classifiers trained with 13 different feature subsets. NF is the number of features in each set.

Pareto Point	**NF**	**LDA**		**QDA**		**SVM-Linear**		**SVM-RBF**		**MLP**		**DT**	
		ACC	STD	ACC	STD	ACC	STD	ACC	STD	ACC	STD	ACC	STD
p_1	6	93.56	0.740	94.33	0.852	95.37	1.218	98.33	0.421	97.34	0.698	96.15	1.16
p_2	7	95.31	0.829	96.06	0.711	96.99	0.775	98.88	0.216	98.29	0.539	96.35	1.00
p_3	8	96.69	0.835	96.82	0.410	97.47	0.694	98.86	0.378	98.20	0.411	96.67	1.18
p_4	9	96.86	0.684	96.95	0.484	98.07	0.576	99.31	0.234	98.34	0.459	96.73	1.25
p_5	10	97.04	0.657	96.99	0.427	98.08	0.430	98.90	0.406	98.47	0.645	96.56	1.22
p_6	11	96.84	0.536	97.15	0.654	98.14	0.692	98.94	0.293	98.62	0.578	96.32	1.28
p_7	12	96.93	0.656	97.36	0.372	98.20	0.497	99.05	0.356	98.87	0.406	97.21	1.00
p_8	13	96.61	0.384	97.62	0.554	98.25	0.534	99.14	0.305	95.76	9.180	96.86	1.36
p_9	14	96.76	0.485	97.79	0.311	98.49	0.500	98.88	0.290	98.90	0.355	97.15	0.78
p_{10}	16	96.95	0.525	97.84	0.578	98.59	0.467	98.38	0.449	98.66	0.371	96.99	0.72
p_{11}	21	97.13	0.501	97.93	0.716	98.62	0.564	98.40	0.392	99.00	0.432	97.30	0.54
p_{12}	27	97.41	0.568	97.77	0.861	98.70	0.422	97.58	0.663	99.07	0.373	96.91	0.64

Table 7. Comparison of classification performance using Wilcoxon signed-rank tests (W.T.) at a 5% significance level. B or W indicates that the row method performs better or worse than the column method, respectively, while T shows that they tie with similar performance. These results are obtained using all the data from Table 6.

		DT		**SVM-RBF**		**SVM-linear**		**QDA**		**LDA**	
		p-value	W.T.	p-value	W.T.	p-value	W.T.	p-value	W.T.	p-value	W.T.
MLP	vs.	2.44E-4	B	7.32E-1	T	8.50E-3	B	5.02E-3	B	2.44E-4	B
DT	vs.	–		1.23E-4	W	8.20E-3	W	1.33E-1	T	1.70E-1	T
SVM-RBF	vs.	*		–		6.70E-3	B	2.44E-4	B	1.22E-4	B
SVM-linear	vs.	*		*		–		1.15E-4	B	1.25E-4	B
QDA	vs.	*		*		*		–		2.44E-4	B

3.5. Performance Assessment of Selected Subset

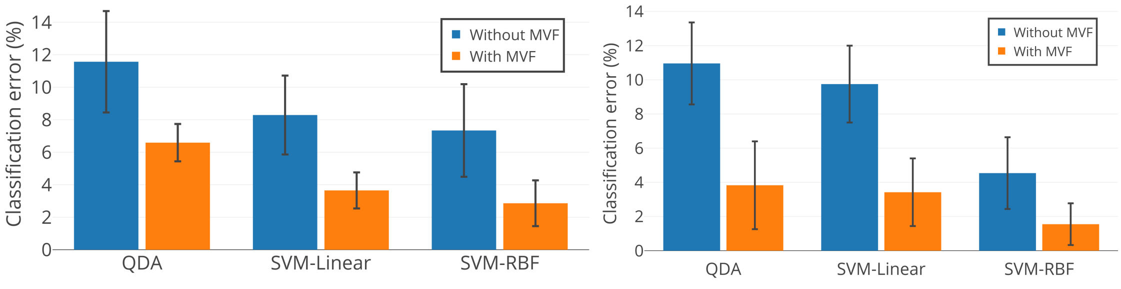
In this section we train UIR for all able-bodied and transfemoral amputee subjects with feature subset p_9. All classifiers are trained with three representative methods (SVM-RBF, SVM-linear, and QDA). The RBF kernel tuning parameter is $\sigma = 1$ and $\sigma = 4$ for able-bodied and amputee subjects, respectively. In this section, we use multiple-fold CV for learning, where each walking sequence is considered a fold.

We saw in Section 2.2 that overlapped windowing with frame length $L_f = 250$ ms and increment $I = 50$ ms is the best data window option. For real-time operation, a conservative choice for parameter $q = 5$ satisfies the constraint $q \times I \leq 300$ ms. Therefore, we use MVF with length $2 \times q + 1 = 11$. Results verify a fast processing time on a standard desktop computer of less than 50 ms, on average, including feature extraction and classification with each of the three classifiers.

Figure 9 illustrates the mean classification error of QDA, SVM-linear, and SVM-RBF trained with feature subset p_9. Training was conducted individually for able-bodied subjects AB01, AB02, and AB03 and amputee subjects AM01, AM02, and AM03, with and without MVF. Figure 9 indicates that: (1) SVM-RBF outperforms SVM-linear and QDA, which confirms the statistical results in Section 3.4; (2) MVF statistically significantly decreases classification error for locomotion mode detection ($p < 0.05$); and (3) p_9 is an effective feature subset and results in accurate as well as compact UIR. Feature subset p_9 uses only 14 features out of a total of 60 available features, which reduces the size of the feature set by 77%.

SVM-RBF was also trained for the able-bodied subjects with the full set of 60 features. When combined with MVF it results in a mean classification accuracy of 98.54% ± 1.92%. In comparison, we achieve 97.14% ± 1.51% mean classification accuracy with feature subset p_9, which includes only 14 features. Statistical tests at 5% significance level indicate no significant difference between UIR performance when trained with the full feature set and subset p_9.

SVM-RBF was also trained for the amputee subjects with the full set of 60 features. When combined with MVF it results in a mean classification accuracy of 99.37% ± 0.96%. In comparison, we achieve 98.45% ± 1.22% mean classification accuracy with feature subset p_9. As with the able-bodied subjects, statistical tests indicate no significant difference between UIR performance when trained with the full feature set and subset p_9. This indicates the satisfactory performance of our framework, which is able to eliminate unneeded features with no significant degradation in overall accuracy.



(a) Trained with subset p_9 for able-bodied subjects

(b) Trained with subset p_9 for amputee subjects

Figure 9. Classification performance of QDA, SVM-Linear, and SVM-RBF with feature subset p_9 for able-bodied subjects (AB01, AB02, and AB03) and amputee subjects (AM01, AM02, and AM03).

4. Conclusion

We presented a framework for designing a UIR system. We used experimental data collected from three able-bodied subjects and three above-knee amputee subjects to classify four and three different gait modes, respectively. Overlapped windowing with frame length 250 ms and increment 50 ms provided a good tradeoff between classification performance and real-time computation. Several efficient TD and FD features were extracted from data frames to form the feature set. We performed feature selection in two steps. First, we excluded non-informative features with poor classification performance and high computational effort. Second, we used MOO to find an optimal feature subset from the remaining features to obtain a UIR system that was both parsimonious and accurate. For this purpose, GMOFS, a novel embedded multi-objective feature selection algorithm, was proposed and compared with four evolutionary MOOs on the basis of normalized hypervolume and relative coverage. Classification results confirmed the competitive performance of GMOFS. Several classifiers were trained with the optimal feature subsets that were selected by MOO, and SVM-RBF and MLP were found to be the best classifiers for UIR. The outputs of the classifiers were input to an MVF to improve classification accuracy and chattering between the identified classes.

For future work, more above-knee amputee subjects will be involved in data collection and classification. In addition, we will include other daily-life activities such as incline walking, stair ascent and descent, standing and sitting, etc. It is also of great interest to consider other informative features for classification, such as wavelet transform coefficients. Finally, it would be of interest to compare GMOFS with other state-of-the-art MOO methods, and to apply GMOFS to other MOO problems.

Author Contributions: Investigation, G.K.; methodology, G.K. and H.M.; project administration, D.S.; resources, H.M.; software, G.K.; writing—original draft preparation, G.K. and H.M.; writing—review editing, D.S.; funding acquisition, D.S.

Funding: This research was supported by National Science Foundation grants 1344954 and 1536035, and a Cleveland State University Graduate Student Research Award.

Acknowledgments: The authors would like to thank Dr. Elizabeth C. Hardin from Cleveland Veterans Affairs Medical Center who contributed to the experiments of collecting data.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript.

List of acronyms in order of appearance

Acronym	Definition	Acronym	Definition
UIR	User intent recognition	ZC	Zero crossing
MOO	Multi-objective optimization	WL	Waveform length
GMOFS	Gradient-based multi-objective feature selection	VAR	Variance
MOBBO	Multi-objective biogeography-based optimization	MAV	Mean absolute value
SVM	Support vector machine	RMS	Root mean square
RBF	Radial basis function	WAMP	Willison amplitude
MVF	Majority voting filter	SK	Skewness
sEMG	Surface electromyography	KU	Kurtosis
LDA	Linear discriminant analysis	COR	Correlation
QDA	Quadratic discriminant analysis	ANG	Angle
GMM	Gaussian mixture model	PSD	Periodogram spectrum density
ANN	Artificial neural network	MNF	Mean frequency
BBO	Biogeography-based optimization	MDF	Median frequency
VEBBO	Vector evaluated BBO	MAXF	Maximum frequency
NSBBO	Non-dominated sorting BBO	AR	Auto-regressive model
NPBBO	Niched Pareto BBO	CV	Cross validation
SPBBO	Strength Pareto BBO	AB01	Able-bodied subject 01
EA	Evolutionary algorithm	AM01	Amputee subject 01
MLP	Multilayer perceptron	PS	Preferred speed
TD	Time domain	ST	Standing
FD	Frequency domain	NW	Normal walking
FLDA	Fisher's linear discriminant analysis	SW	Slow walking
PCA	Principal component analysis	FW	Fast walking
DT	Decision tree		
SSC	Slope sign change		

References

1. Lawson, B.E.; Varol, H.A.; Huff, A.; Erdemir, E.; Goldfarb, M. Control of stair ascent and descent with a powered transfemoral prosthesis. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* **2013**, *21*, 466–473.
2. Khademi, G.; Richter, H.; Simon, D. Multi-objective optimization of tracking/impedance control for a prosthetic leg with energy regeneration. IEEE Conference on Decision and Control, 2016, pp. 5322–5327.
3. Khademi, G.; Mohammadi, H.; Richter, H.; Simon, D. Optimal mixed tracking/impedance control with application to transfemoral prostheses with energy regeneration. *IEEE Transactions on Biomedical Engineering* **2018**, *65*, 894–910.
4. Sup, F.; Varol, H.A.; Mitchell, J.; Withrow, T.J.; Goldfarb, M. Preliminary evaluations of a self-contained anthropomorphic transfemoral prosthesis. *IEEE/ASME Transactions on Mechatronics* **2009**, *14*, 667–676.
5. Tucker, M.R.; Olivier, J.; Pagel, A.; Bleuler, H.; Bouri, M.; Lambercy, O.; del R Millán, J.; Riener, R.; Vallery, H.; Gassert, R. Control strategies for active lower extremity prosthetics and orthotics: a review. *Journal of Neuroengineering and Rehabilitation* **2015**, *12*, 1.
6. Huang, H.; Kuiken, T.A.; Lipschutz, R.D. A strategy for identifying locomotion modes using surface electromyography. *IEEE Transactions on Biomedical Engineering* **2009**, *56*, 65–73.
7. Zhang, F.; Huang, H. Real-time recognition of user intent for neural control of artificial legs. Myoelectric Control/Powered Prosthetics Symposium, 2011.
8. Stolyarov, R.; Burnett, G.; Herr, H. Translational motion tracking of leg joints for enhanced prediction of walking tasks. *IEEE Transactions on Biomedical Engineering* **2018**, *65*, 763–769.
9. Hargrove, L.; Englehart, K.; Hudgins, B. The effect of electrode displacements on pattern recognition based myoelectric control. Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2006, pp. 2203–2206.
10. Winkel, J.; Jorgensen, K. Significance of skin temperature changes in surface electromyography. *European Journal of Applied Physiology and Occupational Physiology* **1991**, *63*, 345–348.
11. Zachariah, S.G.; Saxena, R.; Fergason, J.R.; Sanders, J.E. Shape and volume change in the transtibial residuum over the short term: Preliminary investigation of six subjects. *Journal of Rehabilitation Research and Development* **2004**, *41*, 683.
12. Varol, H.A.; Sup, F.; Goldfarb, M. Multiclass real-time intent recognition of a powered lower limb prosthesis. *IEEE Transactions on Biomedical Engineering* **2010**, *57*, 542–551.
13. Khademi, G.; Mohammadi, H.; Simon, D.; Hardin, E.C. Evolutionary optimization of user intent recognition for transfemoral amputees. IEEE Biomedical Circuits and Systems Conference, 2015, pp. 1–4.
14. Liu, M.; Wang, D.; Huang, H.H. Development of an environment-aware locomotion mode recognition system for powered lower limb prostheses. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* **2016**, *24*, 434–443.
15. Wang, X.; Wang, Q.; Zheng, E.; Wei, K.; Wang, L. A wearable plantar pressure measurement system: Design specifications and first experiments with an amputee. *Intelligent Autonomous Systems* **2013**, pp. 273–281.
16. Huang, H.; Zhang, F.; Hargrove, L.J.; Dou, Z.; Rogers, D.R.; Englehart, K.B. Continuous locomotion-mode identification for prosthetic legs based on neuromuscular–mechanical fusion. *IEEE Transactions on Biomedical Engineering* **2011**, *58*, 2867–2875.
17. Young, A.J.; Simon, A.M.; Fey, N.P.; Hargrove, L.J. Classifying the intent of novel users during human locomotion using powered lower limb prostheses. IEEE International Conference on Neural Engineering, 2013, pp. 311–314.
18. Ha, K.H.; Varol, H.A.; Goldfarb, M. Volitional control of a prosthetic knee using surface electromyography. *IEEE Transactions on Biomedical Engineering* **2011**, *58*, 144–151.
19. Young, A.J.; Hargrove, L.J. A Classification Method for User-Independent Intent Recognition for Transfemoral Amputees Using Powered Lower Limb Prostheses. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* **2016**, *24*, 217–225.
20. Young, A.; Kuiken, T.; Hargrove, L. Analysis of using EMG and mechanical sensors to enhance intent recognition in powered lower limb prostheses. *Journal of Neural Engineering* **2014**, *11*, 056021.
21. Simon, D.; Omran, M.G.; Clerc, M. Linearized biogeography-based optimization with re-initialization and local search. *Information Sciences* **2014**, *267*, 140–157.

doi:10.20944/preprints201811.0094.v1

22. Simon, D. *Evolutionary Optimization Algorithms*; John Wiley & Sons, 2013.
23. Xue, B.; Zhang, M.; Browne, W.N. Particle swarm optimization for feature selection in classification: A multi-objective approach. *IEEE Transactions on Cybernetics* **2013**, *43*, 1656–1671.
24. Ghaemi, M.; Feizi-Derakhshi, M.R. Feature selection using forest optimization algorithm. *Pattern Recognition* **2016**, *60*, 121–129.
25. Hall, M.A.; Smith, L.A. Feature Selection for Machine Learning: Comparing a Correlation-Based Filter Approach to the Wrapper. FLAIRS Conference, 1999, Vol. 1999, pp. 235–239.
26. Kohavi, R.; John, G.H. Wrappers for feature subset selection. *Artificial Intelligence* **1997**, *97*, 273–324.
27. Saeys, Y.; Inza, I.; Larrañaga, P. A review of feature selection techniques in bioinformatics. *Bioinformatics* **2007**, *23*, 2507–2517.
28. Ma, S.; Huang, J. Penalized feature selection and classification in bioinformatics. *Briefings in Bioinformatics* **2008**, *9*, 392–403.
29. Guyon, I.; Weston, J.; Barnhill, S.; Vapnik, V. Gene selection for cancer classification using support vector machines. *Machine Learning* **2002**, *46*, 389–422.
30. Hoerl, A.E.; Kennard, R.W. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics* **1970**, *12*, 55–67.
31. Tibshirani, R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* **1996**, pp. 267–288.
32. Zou, H.; Hastie, T. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **2005**, *67*, 301–320.
33. Ji, T. Frequency and velocity of people walking. *Structural Engineer* **2005**, *84*, 36–40.
34. van den Bogert, A.J.; Geijtenbeek, T.; Even-Zohar, O.; Steenbrink, F.; Hardin, E.C. A real-time system for biomechanical analysis of human movement and muscle function. *Medical and Biological Engineering and Computing* **2013**, *51*, 1069–1077.
35. Oskoei, M.A.; Hu, H. Support vector machine-based classification scheme for myoelectric control applied to upper limb. *IEEE Transactions on Biomedical Engineering* **2008**, *55*, 1956–1965.
36. Hudgins, B.; Parker, P.; Scott, R.N. A new strategy for multifunction myoelectric control. *IEEE Transactions on Biomedical Engineering* **1993**, *40*, 82–94.
37. Valls-Solé, J.; Rothwell, J.C.; Goulart, F.; Cossu, G.; Munoz, E. Patterned ballistic movements triggered by a startle in healthy humans. *Journal of Physiology* **1999**, *516*, 931–938.
38. Stark, L. *Neurological Control Systems: Studies in Bioengineering*; Springer Science & Business Media, 2012.
39. Phinyomark, A.; Limsakul, C.; Phukpattaranont, P. A novel feature extraction for robust EMG pattern recognition. *arXiv preprint arXiv:0912.3973* **2009**.
40. Oskoei, M.A.; Hu, H. Myoelectric control systems – A survey. *Biomedical Signal Processing and Control* **2007**, *2*, 275–294.
41. Englehart, K.; Hudgins, B. A robust, real-time control scheme for multifunction myoelectric control. *IEEE Transactions on Biomedical Engineering* **2003**, *50*, 848–854.
42. Guyon, I.; Elisseeff, A. An introduction to variable and feature selection. *Journal of Machine Learning Research* **2003**, *3*, 1157–1182.
43. Oskoei, M.A.; Hu, H. GA-based feature subset selection for myoelectric classification. IEEE International Conference on Robotics and Biomimetics, 2006, pp. 1465–1470.
44. Healthcare Inspection Prosthetic Limb Care in VA Facilities. Technical report, Department of Veterans Affairs Office of Inspector General, 2011.
45. Lara, O.D.; Labrador, M.A. A survey on human activity recognition using wearable sensors. *IEEE Communications Surveys and Tutorials* **2013**, *15*, 1192–1209.
46. Moore, B. Principal component analysis in linear systems: Controllability, observability, and model reduction. *IEEE Transactions on Automatic Control* **1981**, *26*, 17–32.
47. Johnson, R.A.; Wichern, D.W. *Applied Multivariate Statistical Analysis*; Prentice Hall, Upper Saddle River, 2002.