MPTrain: A Mobile, Music and Physiology-Based Personal Trainer

Nuria Oliver[1] & Fernando Flores-Mangas
Microsoft Research, Redmond, WA 98052 USA
[1]nuria@microsoft.com

March 2006

Technical Report
MSR-TR-2006-31

Microsoft Research
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052
http://www.research.microsoft.com

1 Introduction and Previous Work

The influence of music in exercise performance has intrigued the research community for quite some time, leading to numerous research studies on the topic [2, 7, 8, 3, 1, 12]. The overwhelming majority of previous research points to very positive effects of music while exercising. A few of the reasons proposed include that music provides a pacing advantage, a form of distraction from the fatigue of exercising, affects the mood in a positive way, raises confidence and self-esteem and motivates users to exercise more. Finally, ten different studies agree that exercise endurance, performance perception and perceived exertion levels are positively influenced by music versus non-music conditions [13].

It is no surprise, therefore, that music is often part of the exercise routine of many teens and adults. In particular, MP3 players and heart-rate monitors [4, 16] are becoming increasingly pervasive when exercising, especially when walking, running or jogging outdoors. It is not uncommon in the running community to prepare a "running music playlist" [5] that seems to help runners in their training schedules. Recently, interval runner Jeff Welch developed a script which creates an iTunes playlist in which songs stop and start at time intervals to indicate when to switch from running to walking without having to check a watch [14].

However, none of the existing systems to date directly exploits the effects of music on physiology and physical activity in an adaptive and real-time manner. During our background research, we found that all the systems and prototypes developed so far operate in a one-way fashion. They deliver a pre-selected set of songs in a specific order. In some cases, they might independently monitor the user's heart-rate, but do not include feedback about the user's state or performance to affect the music selection. The MPTrain system presented in this paper addresses these limitations.

MPTrain is a mobile phone based system that takes advantage of the influence of music in exercise performance enabling users to more easily achieve their exercise goals. MPTrain is designed as a mobile and personal system (hardware and software) that users wear while exercising (walking, jogging or running). MPTrain's hardware includes a set of physiological sensors wirelessly connected to a small personal computer (*e.g.* cell phone, PDA, etc.) carried by the user. MPTrain's software allows the user to enter a desired workout in terms of desired heart-rate stress over time. It then assists the user in achieving the desired exercising goals by: (1) constantly monitoring his/her physiology (heart-rate in number of beats per minute) and movement (speed in number of steps per minute); and (2) selecting and playing music (MP3s) with specific features that will guide him/her towards achieving the desired exercising goals. MPTrain uses algorithms that learn the mapping between musical features (*e.g.* volume, beat and energy), the user's current exercise level (*e.g.* running speed or gait) and the user's current physiological response (*e.g.* heart-rate). The goal is to automatically choose and play the "right" music to encourage the user to speed up, slow down or maintain the pace while keeping him/her on track with the desired workout.

The paper is structured as follows: In Section 2 we present MPTrain's system architecture. Section 3 is devoted to describing MPTrain's hardware. The signal processing and music selection algorithms are presented in Section 4. MPTrain's user interface is described in Section 5. Preliminary experimental results are presented in Section 6. Finally, our future directions of research are outlined in Section 7.

2 Architecture

Figure 1 depicts MPTrain's architecture. The left side of the Figure, shows a block diagram of the *sensing module*, whose main components are: (1) a set of physiological and environmental sensors (ECG, accelerometers, etc), (2) a processing board that receives the raw sensor signals, digitizes them and passes them to a (3) Bluetooth transmitter, which sends the data wirelessly to the Mobile Computing Device (cell phone, PDA, etc.).

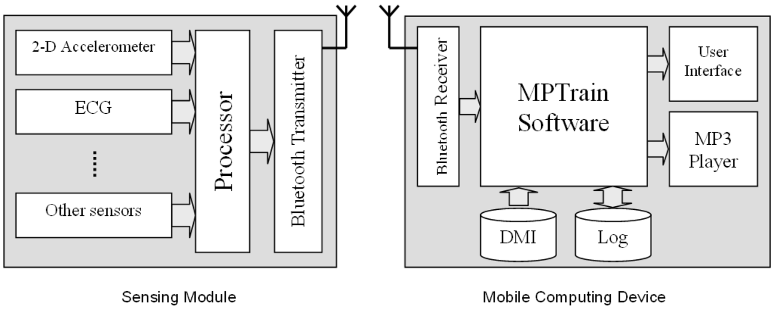

Figure 1: MPTrain's Architecture.

The right side of the Figure depicts is a block diagram of the *mobile computing module* and the components within that are relevant to the MPTrain system: (1) The Bluetooth receiver receives the sensed data and makes it available to MPTrain's software; (2) MPTrain's software receives the raw physiological and acceleration data and extracts heart-rate and gait or speed from it. It accesses the music from the user's Digital Music Library (DML). It logs the user's current session (heart-rate, speed and song being played), and automatically determines the music that needs to be played in order to assist the user in achieving his/her predefined exercise goals; (3) MPTrain has a graphical user interface that shows the current state (*e.g.* heart-rate, gait, song played, location within the current exercise program, volume, etc.), and allows users to (a) enter their desired exercise pattern, (b) set parameters of operation, (c) change songs if desired and (d) enter their personal data, including their resting and maximum allowed heart-rates; (4) Finally, MPTrain interfaces with an MP3 player that reproduces the songs selected by MPTrain at the required speed.

Figure 2 illustrates MPTrain's data flow. The user is listening to digital music in his/her mobile phone while jogging. At the same time, the user's hear-rate and speed are being monitored and logged in the mobile phone. A few seconds (typically 10 s) before the conclusion of the song currently being played, MPTrain compares the user's current heart-rate with the desired one, according to the current pre-selected workout. The *next action module* determines if the user needs to speed up, slow down or keep the pace of jogging, based on whether his/her heart-rate needs to increase, decrease or keep the same pace. With this information, the *music finding module* identifies the next song to be played from the music database. The current implementation of the music finding algorithm determines the next song by identifying a song (1) that hasn't been played yet; and (2) has a tempo (in beats per minute) that is similar to the user's current gait, but increasing or decreasing an amount inversely related to the deviation between the actual heart-rate and the desired heart-rate from the preset workout.

At any instant of time, the user can check how well (s)he is doing with respect to the desired exercise level, modify the exercising goals and change the music track from the one automatically selected by MPTrain. We are currently working on including additional user adaptation to MPTrain, by incorporating information about that user's past sessions and by constantly monitoring the user's actions and learning from them. As the user interacts with the MPTrain system, we would like for its music selection algorithm to become progressively better suited for that particular user.

Next, we shall describe MPTrain's hardware and software components in some detail.

3 Hardware

The current implementation of MPTrain consists of the following hardware devices, as illustrated in Figure 3:

1. Alivetec Alive heart monitor (right in the Figure). It is a light-weight (60 g with battery), low-power (60 hours of operation with continuous wireless trasnmission) wearable sensing device. It has a single channel electrocardiogram (ECG) with 2 electrodes (300 samples (8 bit) per second), a 2-axis accelerometer (75 samples (8 bit) per second), an event button, a Secure Digital card for local storage and a Bluetooth class 1 (up to 100 m range) transmitter that sends its data to the mobile phone, using a Serial Port Profile, client connection. It has very efficient power management that allows for continuous monitoring for up to one week.

2. Audiovox SMT5600 GSM mobile phone (left in the Figure) running Microsoft's Windows Mobile 2003 operating system. It has built-in support for Bluetooth, 32 MB of RAM, 64 MB of ROM, a 200 MHz ARM processor and about 5 days of stand-by battery life.

The complete MPTrain system runs in real-time, uninterruptedly for about 6 hours before needing to recharge the mobile phone batteries. The largest consumer of power in the current version of the system is the MP3 player[6].

4 Signal Processing and Music Selection Algorithms

The sensing module in MPTrain provides raw ECG and acceleration data. The data is continously sent to the mobile phone via Bluetooth. From this raw data stream, MPTrain determines the user's current heart-rate (in beats per minute) and running gait (in steps per minute).

4.1 Heart-rate Computation from ECG

An ECG is a graphic record of the heart's electrical activity. This non-invasive measure is typically obtained by positioning electrical sensing leads (electrodes) on the body in standardized locations. In MPTrain, we use a 2-lead ECG that is positioned on the user's torso, either via a chestband or with 2 adhesive electrodes. MPTrain determines the user's heart-rate from the raw ECG signal by means of the heart-rate detection algorithm described below.

Figure 4 (top) illustrates a typical ECG signal together with the results of the heart-rate detection algorithm. The main steps of the algorithm are as follows:

1. The raw ECG signal (in dark blue in the Figure) is first low-pass filtered to obtain a *ECGLowPassSignal*.

2. Its high-frequency component, named *ECGHighfreqSignal*, is computed by subtracting the *ECGLowPassSignal* from the raw ECG signal.

3. A high-frequency envelope, named *ECGHighFreqEnv*, is computed by low-pass filtering the *ECGHighFreqSignal*.

4. A dynamic threshold, named *ECGThreshold* (in magenta in the Figure), is determined by applying a low-pass filter with very low pass frequency to the *ECGHighFreqEnv*. If *ECGHighFreqEnv*> K∗*ECGThreshold* (where K is typically 3), and a beat has not been detected in the past N samples (where N is typically 10), then a beat is detected (in yellow in the Figure).

5. The user's instantaneous heart-rate (in yellow in the Figure) is computed as:
$$HR_i = (int)\frac{60.0 * SamplingRate}{\#SamplesBetweenBeats} \quad (1)$$
with $30 < HR_i < 300$, and where the *SamplingRate* is $300Hz$ and $\#SamplesBetweenBeats$ is the number of ECG samples since the last detected beat.

6. Finally, a median-filter is applied to HR_i to compute the final heart-rate HR. Median filtering [10] is one of the most common non-linear techniques used in signal processing. It has some nice properties, such as being very robust, preserving edges and removing impulses and outliers.

4.2 Pedometry Computation from Acceleration

MPTrain uses a 2-axis (X and Y) accelerometer. In order to determine the number of steps (pedometry) that the user is taking per minute (SPM), MPTrain's current algorithm only relies on vertical acceleration (Y-acceleration) data. Figure 4 (bottom) depicts a typical raw Y-acceleration signal (in dark blue in the Figure), together with the results of MPTrain's step detection algorithm. Each of the peaks of the blue signal in the Figure corresponds to a step. Thus, the goal of the algorithm is to accurately identify the peaks in the signal.

MPTrain uses a very efficient algorithm that operates in the time domain and is very similar in nature to the heart-rate detection algorithm. First, a low-pass filter is applied to the raw Y-acceleration signal to obtain the *AccLowPassSignal*. Another low-pass filter of much lower pass frequency is applied to the same raw Y-acceleration signal to provide the *AccThreshold* (in magenta in the Figure). The *AccLowPassSignal* is then compared to the *AccThreshold*. Only if *AccLowPassSignal* is lower than the *AccThreshold*, and the acceleration signal has not had a valley yet, a step is detected (in cyan). The instantaneous number of steps per minute is given by:

$$SPM_i = (int)\frac{60.0 * SamplingRate}{\#SamplesSinceLastStep} \quad (2)$$

where the $SamplingRate$ for the acceleration signal is 75 Hz and $\#SamplesSinceLastStep$ is the total number of samples since the last detected step.

Finally, a median filter is applied to SPM_i to obtain the final number of steps per minute, SPM. We ran a set of experiments to determine the accuracy of MPTrain's heart-rate and pedometry detection algorithms. In our experiments, the performance of our algorithms was comparable to that of commercial systems, such as the Polar heart-rate monitor [11] or standard acceleration-based pedometers [9].

4.3 Music Selection Algorithm

MPTrain acts as a personal trainer that uses music to encourage the user to accelerate, decelerate or keep the running speed. The key element is that music improves gait regularity due to the use of the beat, which helps individuals to anticipate the desired rate of movement [15]. The rhythmic structure of the music and the rhythmic actions performed by the body are believed to combine and get sychronized. We shall describe next MPTrain's algorithm for selecting the music to play.

In its current implementation, MPTrain does not take any action until about 10 seconds before the end of the current song. It then determines whether the user needs to increase, decrease or keep the running pace, by comparing the user's current heart-rate with the desired heart-rate from the *desired workout* for that day. Once it has determined the action to take, it searches the user's digital music library (DML) for the optimal song to play. Note that the DML contains not only the user's personal collection of MP3 songs, but also additional information about each song, such as its beat[1] and average energy. Depending on the situation, MPTrain will look for a song whose beat is similar, higher or lower than that of the song currently being played, according to the difference between the actual and desired heart-rates.

For example, if the user's current heart-rate is at 55% of the maximum heart-rate, but the desired heart-rate at that point is at 65%, MPTrain will find a song that has faster beat than the one currently being played. MPTrain respects the user's physical limitations by selecting a song with a beat slightly higher (within a $15-20\%$ range) than the current one.

MPTrain's current implementation uses an empirically learned function to map the physiological response to the music's beat. This model is used to make a statistically accurate track selection. Further versions will also incorporate information about the user's past performance and specific response to each song.

5 User's Interface

MPTrain's software is implemented as a Windows Mobile application, with all its modules (sensor data reception, data analysis, display, storage and music se-

[1] Determined automatically or manually.

lection and playback) running simultaneously in real-time on the mobile phone.

Figure 5 shows some screenshots of MPTrain's interface. On the left, there is MPTrain's main interface window. The solid graph in the center of the window depicts the *desired workout* pattern for that day. It consists of a graph of the desired workout heart-rate (y-axis) –as % of the user's maximum allowed heart-rate– over time (x-axis). In the Figure, the depicted workout contains a *warm-up* period (left-most part of the graph), with desired heart-rate at 35% of the maximum heart-rate, followed by successively more intense running periods (desired heart-rates at $80, 85$ and 90% of the maximum heart-rate) and ended by a *cool-down* phase (right-most part of the graph), with desired heart-rate at 40% of the maximum heart-rate. When MPTrain is in operation, a line graph is superimposed to the desired workout depicting the user's actual performance. This allows the user to compare in real-time his/her performance to the desired one. The user can easily specify his/her desired workout by either selecting one of the pre-defined workouts or creating a new one (as a simple text file). At the bottom of the Figure there is the name of the song currently being played, the total time of workout and the amount of time that the current song has been playing for. Finally, on the top of the Figure, there is the % of battery life left on the sensing module, the user's current speed, the total number of steps and the user's current heart-rate.

The right image on the Figure depicts the interface window with some of the user's personal information, including the user's name, and his/her resting and maximum allowed heart-rates.

6 Experimental Results

We are currently carrying out a comprehensive user study with runners. The study will last three running sessions. In each of the sessions the runners will know what is the desired exercise pattern for the session and they will have their heart-rate and speed being monitored. The first running session will be without music; the second one with random music from the digital music library that is stored in the phone; and the third one with the MPTrain system. With the study we plan to evaluate, for each of the sessions: (1) how well each runner achieves the pre-defined exercise goal; (2) their perception of the workout; (3) their experience using MPTrain.

Before deploying the comprehensive user study, we have extensively tested MPTrain with one runner. The results have been very positive and encouraging. The focus of these suite of tests has been to evaluate and refine MPTrain's algorithms. In all tests, the Digital Music Library contained 30 songs with durations and tempos ranging from $2:48$ to $5:55$ minutes and 65 to 185 beats per minute, respectively. The songs belonged to a variety of music genres (*e.g.* pop, rock, soul, hip-hop, etc), both instrumental and vocal. Note that MPTrain's metadata about each song includes both the tempo of the song in 20 s window intervals and the average tempo. The music selection algorithm described in Section 4.3 uses the average tempo. Figure 6 depicts a histogram of the average tempo of the songs in the DML.

Figure 7 (top) depicts a summary of one exemplary running session of 42 minutes on a treadmill while using MPTrain. Please note that the ideal scenario is when users are exercising outdoors. However, we have carried most of our tests on a treadmill, because this enables us to easily run experiments and refine the algorithms. The results that we have obtained so far when running outdoors are comparable to those on the treadmill. The main difference is that the *desired workout* has been introduced in the treadmill instead of in MPTrain, and MPTrain selects the songs that will match the user's current speed, as measured by MPTrain.

In the Figure, there are 4 important milestones that have been highlighted during the 42 minute workout. The X axis in the graph depicts time (in number of samples), whereas the Y axis depicts: (a) The tempo of the song that is being played at that instant of time (cyan); (b) the user's current speed in steps per minute (SPM), as measured by MPTrain (magenta); (c) the user's current heart-rate, as measured by MPTrain (dark blue); and (d) the number of the song being played (yellow). The highlighted sections correspond to the following events:

1. After the first song being played is finished, MPTrain selects a song whose tempo matches the user's current running speed.

2. There is an increase in the treadmill speed. MPTrain selects the next song to match the increased gate in the user. Note how the user's heart-rate increases as well, due to an increased running speed.

3. MPTrain continues selecting the optimal song, given the user's current speed. As MPTrain does not play the same song twice, it selects the best song to play from the pool of songs in the DML that haven't been played yet.

4. Finally, the cool down period starts. The user's speed slows down and consequently does the tempo of the songs played during this time interval.

Moreover and for all tests, the running experience was (1) significantly more pleasant with music than without music; (2) with the MPTrain system than with random music; and (3) our subject run harder and felt less tired when listening to music.

Finally, in the outdoor running experiments that we have carried out so far, the *only* case in which the user was able to achieve the desired workout was when using MPTrain. Figure 7 (bottom) depicts typical results when running outdoors while using MPTrain.

The graphs summarize a number of relevant system features. The top graph shows the actual heart-rate (in blue) and the desired heart-rate (in red). The correlation between both is over 0.86, showing that MPTrain certainly drives the user along the preset workout. The bottom graph shows the user's running speed (in blue) and the music tempo (in red). The three highlighted regions correspond to the following events:

- Region **(a)** shows MPTrain's adaptation period at the beginning of workout. As the system knows nothing about the user's current state, it randomly selects a track from the DML and plays it. It also starts data collection for later use. Note how, immediately after region (a) is over, the difference between the user's desired and actual heart-rates is extremely small. Therefore, MPTrain selects a song whose tempo is synchronized with the user's pace.

- Region **(b)** illustrates significant changes in music tempo to synchronize with the user. These changes in music tempo lead to correspondent changes in the user's running speed. Note that the user may not exactly synchronize his/her pace to the music's tempo. However, as MPTrain plays faster beat tracks, the user's pace speeds up, achieving the desired effect of increasing his/her heart-rate.

- Finally, highlighted region **(c)** corresponds to the cool down stage in the workout where once again, the correlation between music tempo and pace can be seen. As the pace slows down, so does the user's heart-rate, matching the desired heart-rate for this region.

7 Conclusions and Future Work

We have presented MPTrain, a mobile phone based system that takes advantage of the influence of music in exercise performance, to enable users to more easily achieve their exercise goals. MPTrain is designed as a mobile and personal system (hardware and software) that users wear while exercising (walking, jogging or running). It consists of a set of physiological sensors wirelessly connected to a mobile phone carried by the user. MPTrain allows the user to enter a desired exercise pattern (in terms of desired heart-rate over time) and assists the user in achieving his/her exercising goals by: (1) constantly monitoring the user's physiology (heart-rate in number of beats per minute) and movement (speed in number of steps per minute); and (2) selecting and playing music with specific features that will cause the user to speed up, slow down or keep the pace to be on track with his/her exercise goals.

We have described the hardware and software components of the MPTrain system, and presented some preliminary results when using MPTrain while jogging.

In addition to carrying out the user study with runners, there are several lines of future research that we would like to pursue with the MPTrain system:

1. We would like to enrich the user's model by adding information about the user's past workouts to enhance the music selection algorithm. We are interested in incorporating information about the user's past performance, typical achieved heart-rates and speeds for each given desired workout, and his/her specific response to each song from the DML. Finally, we are also adding "rating" functionality to MPTrain's interface, such that users can

very easily rate each song with respect to: (a) its effectiveness in the user's workout and (b) how much the user likes listening to it.

2. We are working on incorporating new musical features to the music selection algorithm, such as the song's average energy and the volume at which is being played.

3. MPTrain's current implementation does not alter the speed or volume with which each song is being played, even though it has the capability to do so. We are currently working on dynamically modifying the speed and volume of playing the songs, in addition to automatically selecting the songs.

4. We are interested in incorporating additional contextual information, such as GPS data, body and external temperature, barometric pressure to determine incline, etc. MPTrain will use this information to produce better music selections.

5. We are looking into various approaches to have users share their workout information (both in real-time and historic summaries) with friends and family. There are a number of interesting scenarios and new applications and services that we are considering in this direction.

6. Finally, we are working on different user interfaces to allow users to rate their workout, review their past workouts, and identify trends and deviations from those trends. We would also like to include lifestyle variables, such as diet, overall mood, stress levels, date of the workout, weather conditions, etc, and find correlations between them and the quality of the workouts.

References

[1] M. Anshel and D. Marisi. Effects of music and rhythm on physical performance. *Research Quaterly*, 49:109–112, 1979.

[2] N. Becker, S. Brett, C. Chambliss, K. Crowers, P. Haring, C. Marsh, and R. Montemayor. Mellow and frenetic antecedent music during athletic performance of children, adults, and seniors. *Percept. Motor Skills*, 79(2):pp. 1043–1046, 1994.

[3] A. Beckett. The effects of music on exercise as determined by physiological recovery heart rate and distance. *J. of Music Therapy*, 27:126–136, 1990.

[4] http://ipod.com, 2006.

[5] http://jogtunes.com, 2006.

[6] http://www.fmod.org, 2006.

[7] L. Kravitz. The effects of music on exercise? *IDEA Today*, 12(9):pp. 56–61, 1994.

[8] W. Lee. The effect of music on walking performance of older adults. Master's thesis, Ball State University, 2001.

[9] E. Melanson, J. Knoll, M. Bell, W. Donahoo, J. Hill, L. Nysse, L. Lanningham-Foster, J. Peters, and J. Levine. Commercially available pedometers: considerations for accurate step counting. *Preventive Medicine*, 39(2):361–68, 2004.

[10] I. Pitas and A. Venetsanopoulos. *Nonlinear Digital Filters: Principles and Applications*. Kluwer Academic, 1990.

[11] Polar. Polar watches. *http://www.polarusa.com.*

[12] J. Potteiger, J. Schroeder, and K. Goff. Influence of music on ratings of perceived exertion during 20 minutes of moderate intensity exercise. *Perceptual Motor Skills*, 91:848–854, 2000.

[13] T. Pujol and A. Langenfeld. Influences of music on wingate anaerobic test performance. *Perceptual Motor Skills*, 88(1):292–296, 1999.

[14] I. running iTunes playlist. http://www.lifehacker.com/software/running/interval-running-itunes-playlist-154491.php, 2006.

[15] M. Staum. Music and rhythmic stimuli in the rehabilitation of gait disorders. *Journal of Music Therapy*, 20:69–87, 1983.

[16] Suunto. T6, foot pod, n6hr. *http://www.suunto.com.*

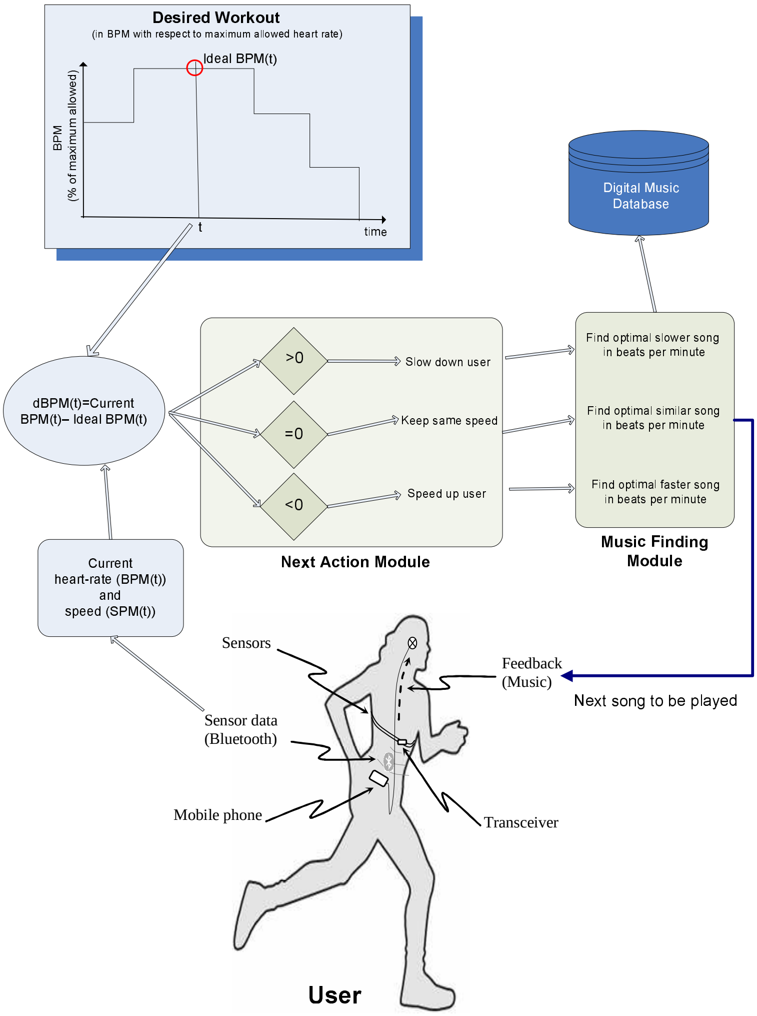

Figure 2: MPTrain's dataflow.

Figure 3: MPTrain's Hardware: On the left, the mobile phone (Audiovox SMT5600) and on the right, the Alivetec heart monitor and 2-axis accelerometer with Bluetooth.

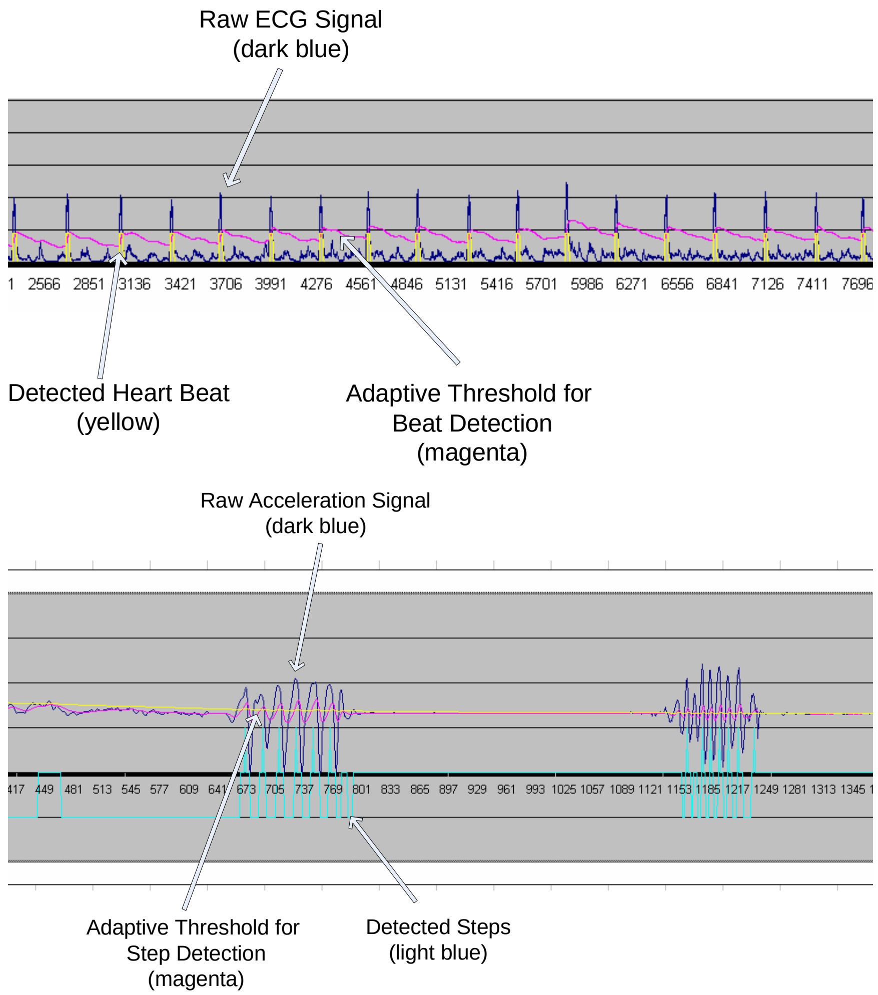

Figure 4: Top: Heart-rate computation (yellow) from raw ECG signal (blue). Bottom: Speed computation (cyan) from raw Y-acceleration (dark blue).

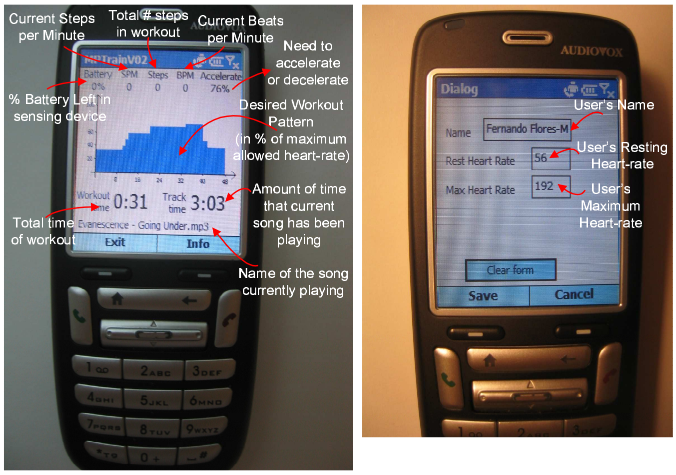

Figure 5: MPTrain's User Interface: On the left, there is the main window with current heart-rate, speed, song being played and desired workout pattern. On the right, there is the user's profile window, with the user's name, with the minimum and maximum allowed heart-rate.

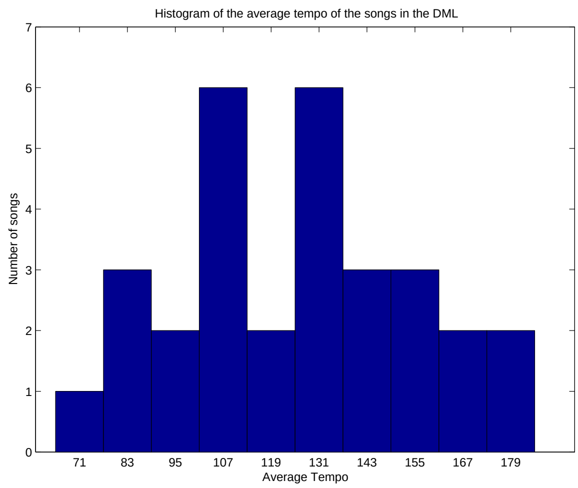

Figure 6: Histogram of the average tempo of the songs in MPTrain's Music Digital Library.

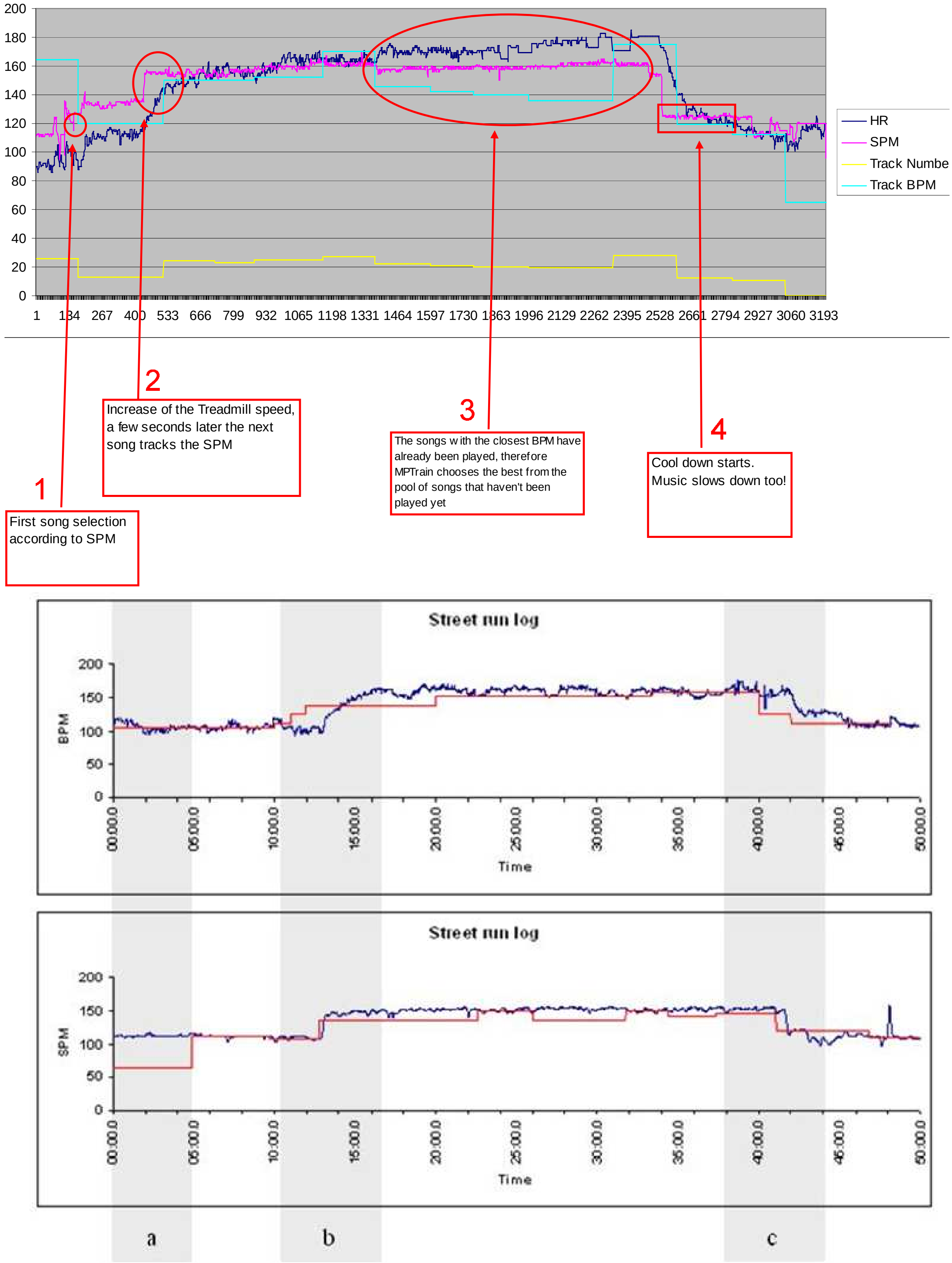

Figure 7: *Top Graph*: 42 minute run log on the Treadmill. The X axis depicts time, whereas the Y axis depicts: (a) tempo of the song that is being played (cyan); (b) user's current speed in steps per minute (magenta); (c) user's current heart-rate (dark blue); (d) number of the song that is being played (yellow). Note how MPTrain selects in all cases songs whose tempo matches as close as possible the users's pace. *Botton Graph*: 50 minute run log outdoors. Again, the X axis depicts time, whereas (a) the top graph depicts the desired heart-rate (red) and real-heart rate (blue); (b) the bottom graph shows the music's tempo (from automatic music selection using MPTrain) in red, and the runner's pace in blue.