Introduction

Problem: Pedometers are widely used in sports settings. We aim to use the MMA8451Q accelerometer on the FRDM KL03 board to implement a pedometer by estimating the time in milliseconds between a wearer's steps, then probabilistically classifying their movement based on their speed. Our categories are 'walking', 'jogging', 'running' and 'none'. This task is challenging due to the KL03's computational constraints (it only has 2KB of RAM), and from capturing the uncertainty in our measurements and algorithm.

Events for Results: walking, jogging, running, or none: all binary events with probability between 0 and 1 inclusive. Let activity $\in$ {walking, jogging, running}. Then, we also have the events 'activity accurately' and 'activity precisely'. P(activity) is a combination of P(activity accurately) and P(activity precisely), and P(none) is a combination of all P(activity), as explained in the Design section.

Relevant Equations: Equation 1 – the x coordinate of the point of intersection between two lines between points (x_{11}, y_{11}) to (x_{12}, y_{12}) and (x_{21}, y_{21}) to (x_{22}, y_{22}) is given by $x = \frac{(y_{21}-y_{11})+\left(\frac{y_{12}-y_{11}}{x_{12}-x_{11}}\right)x_{11}-\left(\frac{y_{22}-y_{21}}{x_{22}-x_{21}}\right)x_{22}}{\left(\frac{y_{12}-y_{11}}{x_{12}-x_{11}}\right)-\left(\frac{y_{22}-y_{21}}{x_{22}-x_{21}}\right)}$.

Equation 2 – for a Gaussian variable $X \sim \mathcal{N}(\mu, \sigma^2)$, $P(a \leq X \leq b) = \Phi\left(\frac{b-\mu}{\sigma}\right) - \Phi\left(\frac{a-\mu}{\sigma}\right)$ where Φ is the CDF.

Equation 3 – for $y = f(x_1, ..., x_n)$ with uncorrelated x_i, $\sigma_y^2 \approx \sum_{i=1}^{n} \left(\frac{\partial f(x_1,...,x_n)}{\partial x_i}\right)^2 \sigma_{x_i}^2$.

Parameters:

Parameter + uncertainty	**Unit**	**Type of uncertainty**
W = duration of the activity measuring window = 10,000	ms	Fixed (None)
N = number of measured steps in the activity measuring window	count	Random + Systematic (A,B)
$\{(t_0, t_1)\}_{1..N}$ = lower (t_0) and upper (t_1) times of positive baseline crossings	ms	Random + Systematic (A,B)
$T_i \pm \sigma_{T_i}$ = time (ms) of step $i \in [1..N]$	ms	Random (A)
$a_i \pm 2.64\%$ = acceleration of stride at time t_i;		
$\sigma = 0.0264$ (sensitivity accuracy in MMA8451Q datasheet)	4096/g	Systematic (B)
$\Delta T \in \{T_j - T_i\}_{\forall i,j} \sim \mathcal{N}(\mu_{\Delta T}, \sigma_{\Delta T}^2)$ = time between strides	ms	Random (A)
$\mu_{\Delta T} \sim \mathcal{N}(\mu_{\Delta T}, \sigma_{\mu_{\Delta T}}^2)$ = mean time between strides $= \frac{1}{N}\sum_{j=1}^{N} \Delta T_j$	ms	Systematic (B)
D = ignore stride duration = 100	ms	Fixed (None)

Design

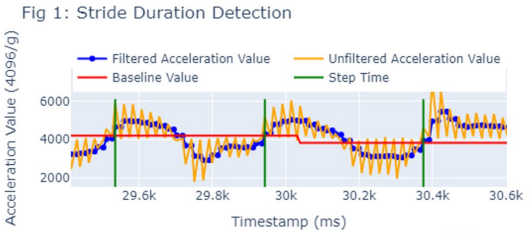
Our Pedometer is based on Zhao's Pedometer[1], which exploits repetitive acceleration patterns when performing a walking-like activity to detect steps. When a person is performing such an activity, their acceleration follows a sharp sinusoidal-looking pattern on the most active axis. By detecting positive gradient crossings over a threshold (baseline), we estimate when a user has stepped (T_i). The time between steps (ΔT) is then estimated and correlated to the activity (ΔT is larger when walking than when running).

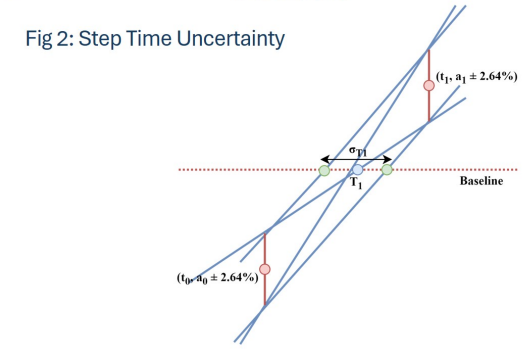
We define the baseline B as half-way between the maximum and minimum value over the previous window W. We focus the acceleration readings and baseline on the most active axis, picked as the axis with the largest difference between its maximum and minimum value over the previous window. To avoid noise being detected as crossings, we filter the signal using a moving average over the previous 4 readings, and ignore subsequent crosses within $D = 100$ms of each other. Given two consecutive readings $a_0 \pm \sigma_a$ and $a_1 \pm \sigma_a$ at times t_0 and t_1 respectively, if outside the ignore period and $a_0 < B$ and $a_1 \geq B$, we have detected a step between t_0 and t_1, as shown in Figure 2. But when specifically? Given the error from the MMA8451Q is $\sigma \approx 0.0264$ and assuming a linear increase between a_0 and a_1, we find all crossings of the lines between (t_0, a_0) and (t_1, a_1) across their error regions that cross the baseline, as shown in Figure 1. By solving the 4 line intersections using Equation 1, we estimate a baseline crossing time $T_1 \pm \sigma_{T_1}$.

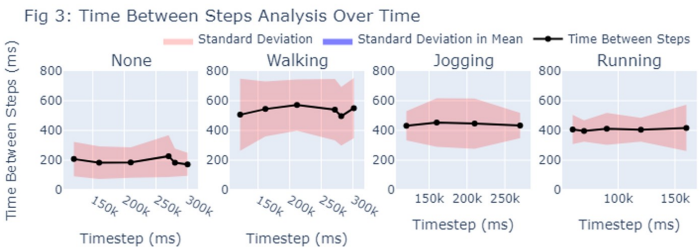
Given uncertain estimates of each step time, we now estimate the time between steps and its uncertainty. Given two baseline crossings $T_0 \pm \sigma_{T_0}$ and $T_1 \pm \sigma_{T_1}$, we estimate the stride duration as $\Delta T_1 = T_1 - T_0 \pm (\sqrt{\sigma_{T_0}^2 + \sigma_{T_1}^2} \approx \sigma_{\Delta T_1})$, propagating the error in T_0 and T_1 via quadrature (using Equation 3). Then, given a sequence of stride duration estimates $\Delta T_i \pm \sigma_{\Delta T_i}$ we compute the mean and variance of ΔT_i to obtain a distribution for ΔT. However, this does not consider each stride duration's error $\sigma_{\Delta T_i}$. We factor this in by propagating their errors via quadrature (Equation 3) to obtain a distribution for $\mu_{\Delta T}$ alongside a distribution for ΔT. This is because $\mu_{\Delta_T} = \frac{1}{N}\sum_{j=1}^{N} \Delta T_j$, thus via quadrature (Equation 3), $\sigma_{\mu_{\Delta_T}}^2 = \frac{1}{N^2}\sum_{j=1}^{N} \sigma_{\Delta T_j}^2$.

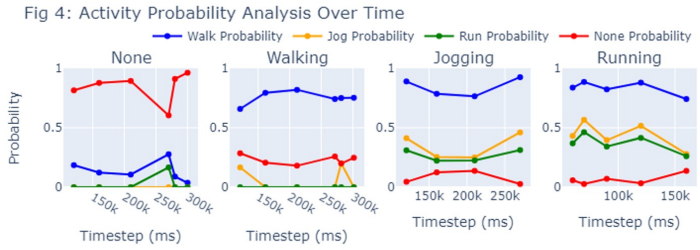
Given distributions for ΔT and $\mu_{\Delta T}$, which factor in random errors and systematic (measurement) errors respectively and representing the distributions of doing the activity accurately and precisely respectively (corresponding

[1] Zhao's Pedometer: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=394cde3eb1aec2987b80140f838e27d4fc76ccd3









to the 'activity accurately' and 'activity precisely' events), we estimate $P(\text{activity})$ for activity $\in$ {walking, jogging, running} as $P(a \leq \Delta T \leq b) * P(u \leq \mu_{\Delta T} \leq v)$ with the specific values of a, b, u, v set to lower and upper bounds for the stride duration of the activity, estimated from a research-backed blog[2] and further tuning. The values are offsets from the expected ΔT for the activity, which for walking, jogging and running we estimate as 545ms, 400ms and 343ms respectively. Our bounds are such that the PDFs overlap between the activities, accounting for the ambiguity between activities. Thus, the probability of each activity does not sum to 1, and we compute $P(\text{none}) = \Pi_{\text{activity}}(1 - P(\text{activity}))$, i.e. the likelihood of doing no activity is the likelihood of not doing each activity in-tern. This assumes independence, among the many other Gaussian assumptions in our design.

Implementation

To implement the algorithm on the KL03, we first gathered test acceleration data and evaluated it in Python for quick prototyping. After obtaining good results, we ported it to the KL03. We removed much of the boot code to free space in the map file and implementd it within `devMMA8451Q.c` in `measureActivityForeverMMA8451Q` which forever reads the accelerations and outputs the estimates of the ΔT distribution parameters and probability of each activity to serial-out every W ms. Every parameter of the algorithm is implemented in an online manner, so it is very space-efficient (we only store the previous 3 accelerations for filtering, and algorithm parameters), and the driver uses only 83 bytes of extra memory. We update the means variances of the ΔT distribution using Welford's algorithm[3]. We update the variance of the $\mu_{\Delta T}$ distribution by formulating its variance $\sigma^2_{\mu_{\Delta T}} = \sigma^2 = \frac{1}{N^2}\sum_{j=1}^{N}\sigma^2_{\Delta T_j}$ as a recurrence relation and updating it recursively: $\sigma^2_{j+1} = \frac{1}{N^2}((N-1)^2\sigma^2_j + \sigma^2_{\Delta T_j})$. For the PDF calculations $P(a \leq \Delta T \leq b)$ and $P(c \leq \mu_{\Delta T} \leq d)$, we formulate them as CDFs of standard Gaussians (Equation 2) and compute the CDFs efficiently using the Abramowitz and Stegun approximation[4]. We also adjust the stored timestamps of the previous step when the KL03 timestamp resets to 0, occuring every 65,536ms due to integer overflow.

Performance Evaluation

To evaluate the performance of our algorithm, we performed the three activities in addition to standing still for 1.5 minutes, and recorded the output estimates of the ΔT distribution in Figure 3 and the activity probabilities $P(\text{activity})$ and $P(\text{none})$ in Figure 4. We observe that, as the intensity of the activity increases, the measured ΔT decreases, validating the algorithm's ability to estimate ΔT. Interestingly, the variance decreases as the activity intensity increases: an unexpected result. From the activity probabilities, we observe that the algorithm successfully distinguishes between no activity and either walking, jogging or running. Also, the algorithm correctly identifies walking with the highest probability when walking, but it is incorrectly the highest probability when jogging and running. However, when jogging and running, the jogging and running probabilities increase, and are larger when running than when jogging, highlighting that the algorithm successfully distinguishes walking, jogging and running. To make the algorithm better distinguish walking from jogging and running, its PDF ranges would need to be reduced for walking, since the PDF range for walking is currently too large and overlaps too many events. Similar tuning would need to be done for jogging and running to more clearly distinguish them.

[2]Pedometer tuning article: https://www.verywellfit.com/pedometer-step-equivalents-for-exercises-and-activities-3435742
[3]Welford's algorithm: https://changyaochen.github.io/welford
[4]Implementation of the Abramowitz and Stegun approximation for the CDF: https://www.johndcook.com/blog/cpp_phi/