Predicting Groundwater Depth Using Handpump Accelerometry Data

Farah Colchester

University College

Supervised by David Clifton, Patrick Thomson and Rob Hope

7th September 2015
Abstract

Safe and reliable access to groundwater is a necessity for people in rural Africa. Many rely on handpumps, for which there is often no system of maintenance in place, leading to around one third being out of use at any given time. Many places are experiencing increased demand for groundwater, leading to the need for a system to monitor groundwater levels.

This paper presents a novel approach to monitoring ground water levels using the “accidental infrastructure” of handpumps. We place accelerometers on the handles of handpumps and record tri-axial data when the pump is in use. The accelerometry data represents both the gross movement and the vibrations of the handle. We show how this data can be used to predict the depth of the water at the pump. We create feature vectors using wavelet-, smoothing spline-, and heteroscedatic Gaussian process-based models of the accelerometry data, and subsequently perform regression using support vector regression or Gaussian process regression. We find that we can obtain low errors of groundwater depth, but that there are biases among the predictions.

In addition, we investigate the effect of the tide on groundwater levels and show how it can be modelled with linear regression.
Chapter 1

Introduction and literature review

1.1 Motivation

Most people in rural sub-Saharan Africa depend on groundwater as their main water supply, accessed through wells or boreholes. Groundwater is generally safe to drink, and hence is a better source than surface water. Traditionally, groundwater is accessed through open wells, but these can easily be contaminated. Many wells are now covered and have a handpump for pumping the water through a borehole. One of the Millennium Development Goals was to halve the proportion of people without access to improved water supplies, which includes covered wells [1]. This goal was the first of the MDGs to be met, and across the developing world the proportion without improved water services has fallen from 30% to 11%. The respective figures for sub-Saharan Africa, however, are 52% to 32%, meaning that there is still substantial progress to be made. However these figures are misleading because they may not take into account cases where the improved water supply is no longer in action. Most rural water schemes rely on the users to maintain them, but this is not effective, and it is estimated that a third of handpumps are out of use at any given time [2].

This particular project is part of a wider collaboration that looks at the operation
and maintenance of rural water supplies. The latter recognises that communities do not have the expertise to maintain handpumps, nor the organisation and methods to collect funds to pay for maintenance, and management needs to be reformed by local authorities or other bodies. The “smart water pumps” study (2013-) has previously attached accelerometers to the water pumps, which measure their use and estimates the volume of water withdrawn from the movement of the handle. This information is transmitted by SMS text message to a central server. The system enables those responsible for the water pumps, such as the local authority or an NGO, to monitor pump use from afar, and to identify which pumps are over-burdened and which are underused; the breakage of pumps can be determined, as can their return to use.

These studies consider handpumps in Kwale County in the south of Kenya. Kwale County borders the coast between the border to Tanzania and the city of Mombasa, and stretches about 100km inland. It has a population of 650,000, of whom 80% live in rural areas. It is one of the poorest parts of the country, with a poverty rate of 74.9%, compared to the Kenyan average of 47.2% [3]. There has been an increase in mining and large-scale agriculture activities in the area, which will greatly increase water consumption. It is important to see what effect this will have on groundwater levels, of concern both to local villagers and for the sustainability of agriculture and industry itself. Currently, information on groundwater levels is very sparse, being measured only at large industrial or agricultural sites. If we could estimate the water level at each pump, it would provide a clearer map of the water level across areas as it changes with time. This more detailed longitudinal water-level data will allow us to, for example, quantify the impact of industry and large-scale agriculture on the local water supply, as well as measure normal
seasonal variations due to rainfall and recharge. This report describes the designs of a novel system for monitoring groundwater level using the “accidental infrastructure” of handpumps. Currently the accelerometers used by the smart water project only collect low-frequency movement of the handle. We aim to replace these with monitors that also capture high-frequency information so that the pump handle vibration is quantified as well as the gross movement caused by pumping. We will use machine learning techniques to predict the depth of the well and hence the groundwater level, as described by this report.

1.2 Water development

Since the 1970s, much effort has been put to improving water supply in rural Africa, both by governments and NGOs. Resources were put into building and installing handpumps. The Rural Water Supply Network (RWSN) is a network of professionals working in rural water supply development. Their study [2] looked at the myths held by those working in this area which have held back development. Resources are often put into building and installing handpumps, with few resources reserved for ongoing maintenance. As well as leading to maintenance problems, this means that far fewer communities can benefit from a given amount of money due to the high costs of hardware. Indeed, one such myth identified is that communities are capable of managing their facilities on their own. A popular concept in water development is “Village level operation and management” (VLOM) [3]. That is, that government or other agencies fund and install handpumps, but it is the responsibility of the users to maintain them. The Afridev handpumps was designed specifically for this method of maintenance; they are robust and simple to
maintain with few separate parts. It is now the second-most common worldwide, after the Indian Mk 2. Under VLOM, local users are given tools and taught how to maintain the pumps on installation, so that they can carry out necessary repairs. However, VLOM has not worked in practice, and is partly responsible for the high failure rate of pumps [5]. Under this system, members of the community are sometimes able to organise and fund minor repairs, but are rarely able to cope with large ones [2]. It is now widely argued that local authorities or NGOs should provide backup support, possibly in association with the private sector [2,4,5]. A major exacerbating problem is lack of information [6]. When pumps are set up by external agents, e.g. local authorities of foreign NGOs, there is no feedback system for information on whether the pump is being maintained and if it is still in operation.

1.3 Smart handpumps

Thomson et al. [7] utilised the increasing mobile coverage in rural sub-Saharan Africa and ever-decreasing cost of electronic parts to reduce this lack of information. They designed a waterpoint data transmitter (WDT) that can be fitted to the handpump and records and transmits usage information. They did a proof of concept trial in Lusaka, Zambia in July, 2011 using a local India Mk II pump. The waterpoint data transmitter was designed so that it could be retrofitted to pumps and not interfere with their normal usage. In addition, it had to be robust, low cost, and scalable. The WDT consists of an accelerometer, microprocessor and GSM transmitter.

The accelerometer filters out high-frequency movement, to record only the gross movement of the handle. The microprocessor takes the accelerometer readings and calculates
the tilt angle of the handle. These angular measurements were sent to the transmitter
and sent by SMS every minute. This data is then used to estimate the amount of water
extracted.

A further study [8] set out to show that the information gap filled by WDTs can
improve water supply for the local people. A region in Kyuso, Kenya, which includes
66 Afridev handpumps was chosen for the study and the pumps had WDTs fitted so
that usage information could be collected. In addition, a trained mechanic was assigned
to fix those pumps when needed, paid by the study rather than the handpump users.
The pumps in half the area were “crowd sourced”, that is, the villagers had to telephone
the mechanic themselves when the pump broke. Pumps in the other half of the area
were actively managed; water extraction was monitored and, when it fell to zero, it was
assumed that the pump was out of order and the mechanic was called. A baseline survey
was done at 21 of the pumps. 18 of the pumps had a breakdown in the previous 12
months, with a median out-of-use time of 6 days and mean of 27, due to some pumps
that were left unfixed for over a year. The methods used to fund pump repairs varied,
but often there were insufficient funds, especially for more major repairs. Under the new
repair model, pump outage time was significantly reduced. For those that were crowd-
sourced the median and mean outage times fell to 3.7 and 3 days respectively, and for
those that were actively managed, it fell to 2 and 1 days. This increase in reliability
of water sources was appreciated by the local population, shown by the fact that most
villagers said that they would be willing to pay for a service of that level.

This system has the potential for great improvements in rural water supply. Firstly,
it would allow authorities to monitor usage of water pumps. If the handpumps in one
location are used heavily, this may indicate that the location could benefit from further handpumps. It would allow those who install the handpumps to know how much it is being used and if it is being maintained. If they realise that many of their pumps are not being maintained, then they can review their processes.

1.4 Water depth measurement and prediction

There are various methods for measuring groundwater depth at wells [9]. The simplest and most common is to “dip” the well, either by dropping chalked tape or wire with an electric sensor which signals when in contact with water. Although cheap, these methods are manually intensive, and hence unfeasible for continuous measurements. Continuous measurements can be taken using a pressure transducer and data logger, but these devices are expensive. Both of these methods require external components which are liable to be damaged over time. They also require that there is access to the well, whereas it is preferable to keep wells sealed to prevent contamination of the water.

Predicting the groundwater level over an area given measurements at certain points is a long-standing problem in geostatistics, for which many methods are used [10]. Common approaches include the more basic variety, such as k-nearest nearest neighbours, where the estimate for an unknown point is simply the average of the measurements of the $k$ nearest points. A commonly used method is kriging [11], a geostatistical technique first proposed by Danie Krige and developed by Matheron [11]. Kriging uses estimated covariance values to interpolate over an area, and can be extended to model over time as well [12]. Machine learning and signal processing methods have also been applied: Wavelets and artificial neural networks (ANNs) have been used to forecast groundwater
levels, as have support vector machines (SVMs).

1.5 Accelerometry data

It would be greatly beneficial in a generally health-monitoring setting to be able to infer a subject’s physical activity during their daily life. To this end, there have been many studies in which human participants wear accelerometers attached to a part of their body, often the torso or leg. Using the resulting accelerometry data studies aim to classify or quantify physical activity. Often broad categories are used to describe activity, such as at rest, low energy motion (e.g., walking) and high energy motion (e.g., running). Various methods are used for feature extraction from the data and classification review the methods. Features extracted from the time domain include the signal magnitude area, and the mean, variance, and skewness of the signal. The Fourier transform can be applied to the signal, and features extracted from the frequency domain, such as the median frequency, total power, or the coefficients of selected frequencies. Wavelet analysis is also used, especially for identifying changes in activity. Many methods are used for classification, including $k$-nearest neighbours, decision trees, artificial neural networks and SVMs. used accelerometry data to identify the behaviour of twelve species of animals.

1.6 Condition monitoring and novelty detection

Condition monitoring and novelty detection is used in a wide range of fields that involve critical systems. By monitoring a feature of the system, abnormal behaviour can be
identified as a sign of upcoming failure, and maintenance can take place before the event. Applications include such varied complex systems as jet engines [18], sensor networks [19] and patient-health monitoring [20].

Novelty detection is used in cases, common in condition monitoring, where the training data consists of many examples of “normal” behaviour, but very few of abnormal behaviour or failures. In these imbalanced cases, standard classification techniques are ineffective. Instead, a model of normality is built using the training data, to which any further data can be compared to identify if it is normal or abnormal. A probabilistic approach can be used, in which a statistical model, such as a Gaussian mixture model, is fitted to the training data. The probability density of further data can calculated assuming the model, and if below a chosen threshold the datapoint is said to be abnormal [21]. If distances can be defined between pairs of points, then a distance-based approach can be used, for example, where data above a certain distance from it nearest neighbours considered an outlier [22]. A further approach uses artificial neural networks trained to replicate the training data; test data that cannot be replicated well can be considered outliers [23].

Gaussian processes and multi-task Gaussian processes have been used in a health setting to model patients’ vital signs in a way that is robust to missing data [24,25]. This multi-dimensional data can be used for novelty detection using multi-variate extreme value theory [26]. A trial of the method for predicting crises in patients in a high-risk hospital ward showed a similar true-positive rate compared to existing methods, with a much reduced false positive-rate.
Chapter 2

Data description

This section outlines the collection of our datasets, the form of the data, and how we model it to predict depth.

2.1 Data

We have two datasets of accelerometry recordings from handpumps. The first, referred to as the “Kenya” dataset was collected in Kwale over a two-week period in 2014, and the second, referred to as the “Kellogg” dataset was collected in Oxford in April - November, 2014. Each data set consists of “recordings” taken at these pumps in the given location. To obtain a recording, a Nintendo Wii remote was strapped to the handle of the pump, around 20cm from end of the handle closest to the shaft. One person pumps water for the whole recording, and the length of recordings ranges from 20s to 120s. The Wii-mote was connected by Bluetooth to a nearby laptop, which recorded the resulting accelerometry measurements. This accelerometry was recorded in three dimensions, and an example is shown in figure 2.1. The Wii-mote records at 96Hz. Figure 2.2 shows an interval of the data in all three dimensions. There is no gross movement in the x-dimension, and a small amount in the z-dimension. The largest range is in the y-dimension, and it is on this that
we perform most of our analysis.

In the y-dimension, we can see the distinctive periodic pumping motions. The decreasing part of the curve with the large amount of “noise” is when the pump handle is being pushed down, and the water is being lifted. The increasing part is when the handles is being lifted up and the water-lifting mechanism is dropping back down. We term each of these motions as a “wave”, and recordings are divided at the troughs to give individual waves.

![Diagram showing the placement of the Wii-mote on the handpump and the direction of accelerometry.](image)

**Figure 2.1**: Diagram showing the placement of the Wii-mote on the handpump and the direction of accelerometry.

The Kenya dataset consists of 85 recordings. These were made at 11 different pumps, of varying depths. We recorded various people pumping - three members of the team from Oxford, one local partner, as well as villagers using the wells. It was not possible to measure the groundwater level at the wells, but an estimate of the water level can be made from the known depth of the well shaft. The data were collected towards the end
of Kenya’s dry season, meaning it is likely that the depth of groundwater was close to the depth of the well. Figure 2.3 shows a histogram of the depth labels of the recordings. Around half of our data came from one well that was around 7m deep, and a further quarter came from wells that were around 40m deep.

A pump was set up in Kellogg College, Oxford. Recordings were made there intermittently, and the “gold standard” depth measurements were taken using a dipper by members of the team. Figure 2.3 shows a histogram of the depths of the recordings. We see that the depths fall into four clusters, and boundaries can be placed at 4.1, 4.2 and 4.4 m.
2.2 Modelling recordings

An outline of the general modelling approach is shown in figure 2.4. We take the 96 Hz recordings, along with their gold standard depth labels. We then model the recordings, that is, we find functions that represent them. Using these, the recordings are divided into individual waves. A proportion of the modelled waves and their labels are used to train a regression system. Finally the “held out” waves are used as test data, and we obtain a prediction for each using the regression system. These predictions can be grouped to give a single prediction of depth for the whole recording.

As described above, we aim to reduce the time-series sensor data to functions that
Figure 2.4: Diagram of the process from time-series sensor data to predictions of depth.

capture their characteristics. There are two main characteristics of the wave: their shape, which represents the gross movement the handle made while pumping, and the “noise” which represents the high-frequency vibrations in the handle during the movement.

There are three candidate methods that we investigate for this purpose:

- The wavelet transform, which is primarily a signal processing method, commonly used to analyse accelerometry data. (Section 2.2.1)

- Smoothing splines, which is a classical statistics method. (Section 2.2.1)

- Heteroscedatic Gaussian processes, which is a machine learning method. (Section 2.2.2 - 2.2.3)

We take the shape of the recording, and separate the time series at the troughs to divide the recordings into the individual waves. The waves are generally between 0.8s and 1.2s in length, and hence contain approximately 80 and 120 data points.
2.2.1 Fourier and wavelet transforms

Signal processing methods are the most widely used in accelerometry analysis, and both Fourier and wavelet transforms are used.

A continuous function can be described as a possibly infinite sum of sine and cosine waves of increasing frequencies, that is, for any real continuous function \( x(t) \)

\[
x(t) = \sum_{n=1}^{\infty} a_n \cos(nt) + \sum_{n=1}^{\infty} b_n \sin(nt)
\]

The values of \(|a_n|\) and \(|b_n|\) are the amplitudes of frequency \( n \), and these give information concerning the frequency components of the signal. The Fourier transform \( \hat{x}(\omega) \) of a function \( x(t) \) gives the amplitude of the frequency components \( \omega \). It is defined as:

\[
\hat{x}(\omega) = \int_{-\infty}^{\infty} x(t) e^{-2\pi i \omega t} dt.
\]

An alternative way to consider \( \hat{x}(\omega) \) is that it is the convolution of \( x(t) \) with cosine and sine waves of frequency \( 2\pi \omega \). If sine and cosine waves of frequency \( 2\pi \omega \) are not present in \( x(t) \) then the convolution of the two, and hence \( x(\omega) \) will be zero. Figure 2.5 shows a Fourier transform applied to an example sequence of 96Hz data. We can see that the frequencies around 1.1Hz have a large amplitude, as this is the frequency of pumping. We see further increases in amplitude at the harmonics of those frequencies.

The Fourier transform provides information about frequency content of a whole recording, but cannot show the amplitude of the frequency components changing through time. One way of addressing this is to use the short term Fourier transform (STFT). This
Figure 2.5: Fourier transform of an example of 96Hz data.

divides the recording into short time windows, and then applies the Fourier transform to each, so that it is possible to see the difference in $\hat{x}$ for each window. However, the resolution of the Fourier transform is fixed, and further resolution in the time domain (i.e., shorter windows) leads to less resolution in the frequency domain; that is, we cannot distinguish as well between different frequencies as window-length decreases in time.

For our recordings, we use a generalisation of the Fourier transform called the wavelet transform. Wavelets are an oscillating function that satisfy certain properties, such as having zero mean and being square integrable. The wavelet transform offers a compromise between the constraints of having high frequency resolution for low frequencies, and high temporal resolution for high frequencies.

From a mother wavelet $\psi(t)$, child wavelets can be formed

$$\psi_{j,k}(t) = \frac{1}{\sqrt{2^j}}\psi\left(\frac{t - k2^j}{2^j}\right)$$

where $j$ is a scale parameter and $k$ is a shift parameter. That is, a first-order child wavelet
is twice the frequency of the mother wavelet, a second-order child wavelet has twice that frequency, and so on.

The coefficient for the lowest frequency $\gamma_{1,1}$ is found by convoluting $x(t)$ with the mother wavelet. This convolution also acts as a high-pass filter, giving a new signal $x_1(t)$. The first and second halves of $x_1(t)$ are then convoluted with the first-order child wavelet, giving two coefficients, $\gamma_{2,1}$ and $\gamma_{2,2}$, representing this frequency in either half of the signal, and each one also acts as a high-pass filter giving new signals $x_2(t)$. This process continues, so that we have $2^n$ coefficients at each level. Specifically, the coefficients are:

$$\gamma_{jk} = \int_{-\infty}^{\infty} x(t) \frac{1}{\sqrt{2^j}} \psi \left( \frac{t - k2^j}{2^j} \right) dt$$

We use the Meyer wavelet as the mother wavelet:

$$\Psi(t) := \begin{cases} 
\frac{1}{\sqrt{2\pi}} \sin \left( \frac{\pi}{2} \nu \left( \frac{3|t|}{4\pi} - 1 \right) \right) e^{jt/2} & \text{if } 2\pi/3 < |t| < 4\pi/3, \\
\frac{1}{\sqrt{2\pi}} \cos \left( \frac{\pi}{2} \nu \left( \frac{3|t|}{4\pi} - 1 \right) \right) e^{jn/2} & \text{if } 4\pi/3 < |t| < 8\pi/3, \\
0 & \text{otherwise},
\end{cases}$$

where

$$\nu(x) := \begin{cases} 
0 & \text{if } x < 0, \\
x & \text{if } 0 < x < 1, \\
1 & \text{if } x > 1.
\end{cases}$$

This wavelet was chosen as it resembles the shape of the waves that we aim to represent.

Figure 2.6 shows the wavelet transform applied to an example of a recording. The first panel shows the 96Hz data and the second panel shows a heatmap representing its wavelet
transform. It can be seen that the steep part of the waves correspond to high amplitudes in the wavelet transform, and the flatter parts to lower amplitudes. The amplitude of these frequencies is much higher than others in the spectrum, and it is clear that these frequencies represent the shape of the wave. We use every 8th frequency between 32 and 128 to create a feature matrix, $s$, defining the “shape” of the recording. We then apply a high-pass filter to the interval, to remove the “shape”, leaving the vibrations, and apply the wavelet transform to the result. We use every 8th frequency between 64 and 384 to create a feature matrix, $v$, representing the detailed vibrations of the recording.

**Figure 2.6:** From top to bottom: a) Interval of recording b) wavelets applied at low frequencies, c) the interval filtered, d) wavelets applied to the filtered interval.
Smoothing splines

Smoothing splines are a solution to the general problem of finding a curve of best fit, \( \hat{f}(t_i) = \hat{f}_i \) through noisy data \( x(t_i) = x_i \). We aim for the square errors \( \epsilon_i^2 = (x_i - \hat{f}_i)^2 \) to be small, but we also want the curve to not be too ‘bumpy’. One measure of the ‘bumpiness’ of \( \hat{f} \) at the point \( t_i \) is the square of its second derivative, \( f'' \): a high value of \( |f''_i| \) suggest the slope of \( \hat{f} \) at \( t_i \) is rapidly changing. A smoothing spline \( \hat{\mu} \) thus compromises between small errors and low bumpiness:

\[
\hat{f} = \arg\min_f \left( \sum (x_i - f_i)^2 + \lambda \int_I f''(t) dt \right)
\]

where \( I \) is the support of the function and \( \lambda \) is a parameter to be chosen. \( \lambda \) can be chosen by cross-validation or inspection, and we chose to use the latter.

To model a wave, we sample the fitted spline at \( p \) intervals thus giving shape \( s \). To obtain a feature representing the noise \( v \) in the wave, we take the mean of \( \epsilon_i \) between each of the \( p \) samples. Figure 2.7 illustrates this.

2.2.2 Gaussian processes

Gaussian processes are a machine learning method for classification and regression. We consider Gaussian processes with reference to univariate time series, but the method extends easily to higher dimensions. A time series of \( n \) points can be thought of as a sample from an \( n \)-dimensional distribution. If the time series is a Gaussian process, it’s joint \( n \)-dimensional distribution is a multi-variate Gaussian. A Gaussian process of data
Figure 2.7: Representation of the smoothing-spline feature vector, for $p = 8$. The second panel shows the spline, with the red circles indicating the values of a feature vector of length $p = 8$. In the third panel the bar chart shows the mean residual from the spline between each sample of the spline.

$x$ observed at times $t$ is defined by a mean function and a covariance function:

$$m(t) = E(x)$$
$$K(t, t') = E((x - m(t))(x' - m(t')))$$

Suppose we wish to predict the noiseless value $f^*$ at point $t^*$. By standard properties of the Gaussian distribution,

$$P(f^*|t^*, t, f) = K(t^*, t)K(t, t)^{-1}f$$

and

$$\text{cov}(f^*|t^*, t, f) = K(t^*, t^*) - K(t^*, t)K(t, t)^{-1}K(t, t^*)$$.

Note that the covariance function only depends on $t$ and not on $f$. 

19
The covariance between two points is proportional to their correlation. Intuitively, we would expect two points close together to be highly correlated, and therefore have high covariance; conversely we would expect two points that are far apart to have low covariance, tending to zero as the distance between them increases. We would also want the covariance between two points to depend just on the distance between them, rather than on their absolute times (i.e., weak stationarity). Hence we chose a covariance function that satisfies these properties. There are several covariance functions that are used; indeed, any positive semi-definite kernel function may be used. One of the most commonly-used is the squared-exponential function

\[ K(t, t') = \sigma^2 f \exp \left( -\frac{(t - t')^2}{2l^2} \right) \]

where \( \sigma \) and \( l \) are hyperparameters corresponding to the \( x \)- and timescale, respectively. \( \sigma \) is the variance of the process; that is, its variation from the latent mean \( f \), and \( l \) represents the time period over which the process could reasonably be extrapolated. We introduce a further noise hyperparameter \( \sigma_n \) where \( x^* \sim N(m(t^*), \sigma_n) \), so that

\[ \text{cov}(t, t^*) = \sigma^2 f \exp \left( -\frac{(t - t')^2}{2l^2} \right) + \sigma_n \delta_{t=t^*} \]

where \( \delta_{t=t^*} = 1 \) if \( t = t^* \) and 0 otherwise. Figure 2.8 shows the effects of varying the values of the hyperparameters.

We use the GPML implementation of Gaussian processes developed by Rasmussen [27]. The training data are presented with their labels, along with starting estimates for the hyperparameters. The covariance matrix \( K \) for the inputed data is calculated using
Figure 2.8: Three Gaussian processes fitted to a 15s interval of data for a woman pumping at a 7m pump. In each case, the blue dots are the data points, the red line is the best fit Gaussian process, and the grey shaded area represents the 95% confidence interval of the GP. \( l, \sigma_f \) and \( \sigma_n \) are respectively: Upper: 0.03, 0.1, 0.01 Middle: 0.21, 0.27, 0.07 Lower: 3.33, 0.91, 0.28

the squared-exponential covariance function and the starting values of the hyperparameters. The negative log marginal likelihood (NLML) of the data given the hyperparameters is then calculated; that is

\[
l(x|t, \theta) = \frac{1}{2} x^T (K + \sigma_n^2 I)^{-1} x + \frac{1}{2} \log |K + \sigma_n^2 I| + \frac{n}{2} \log(2\pi)
\]

We wish to find the value of the hyperparameters that minimise the NLML. To do this, a nonlinear conjugate gradient-descent method is used. This assumes that the likelihood is approximately quadratic around the minimum. The gradient of the likelihood with respect to the hyperparameters is calculated, and the path of steepest descent is found. The minimum of the function along this path is found, using step estimates of the Polak Ribire “flavour” [28]. The process is repeated from this new minimum. This iterates until
we find a local minimum, or reach a maximum number of iterations.

The mean and covariance functions are then calculated using these maximum-likelihood parameter values, and these are known as the posterior predictions. To create the shape feature vector, $s$, we sample the posterior for $f$ at $p$ intervals, and to create the vibrations feature vector, $v$, we sample the posterior for $g$ at $p$ intervals.

### 2.2.3 Heteroscedatic Gaussian Processes

In our formulation of $x(t) = f(t) + \epsilon(t)$, the standard Gaussian processes described above, the mean function describes $f(t)$, but we do not have a function $\epsilon(t)$; it is assumed that $\epsilon \sim N(0, \sigma_n)$ which is independent of time.

Heteroscedastic Gaussian processes (HGPs) assume that residuals $\epsilon_i \sim N(0, \sigma_n(t_i))$, thereby making $\epsilon$ dependent on time. This noise variance $\sigma(t)$ must be positive, and so we take $\sigma(t) = \exp(g(t))$ and model $g(t)$ as a Gaussian process, $g(t) \sim \mathcal{GP}(\sigma_0, k_g(t, t'))$. We assume that the noise is conditionally independent given $t$, and so $k_g$ is the white-noise covariance function, that is, all off-diagonal elements of $k_g = 0$.

We need to calculate the maximum likelihood estimates of the two sets of hyperparameters for the GP over $x$ and $g$. Unlike the standard case, the NLML for a HGP cannot be computed analytically. HGPs were first developed with the posterior distribution calculated using Gibbs sampling [29]. Although this is still the gold-standard method for calculating the posterior noise variance [30], it is very slow. This was first improved by using a “most likely” noise approach for the calculation [31], and later by using variational methods [30]. This was shown to be faster and more accurate than previous methods.

A lower bound for the likelihood is found using a variational approximation. We aim to
find \( P(Y|f, g) \), but instead we approximate it by another function \( Q(f, g) = Q_f(f)Q_g(g) \), where the \( Q_i \) come from a restricted set of distributions. To measure the divergence between \( p(y) \) and \( Q(f, g) \) the Kullback-Leibler divergence is used:

\[
D_{KL}(Q|P) = \sum_Z Q(f, g) \log \frac{Q(f, g)}{p(f, g|y)}.
\]

The lower bound of the likelihood \( L(Q) \) is then defined as

\[
L(Q) = \log p(y) - D_{KL}(Q|p) = \log p(y) - D_{KL}(Q_f Q_g|p)
\]

To remove the dependence on \( Q_f \) we find the distribution \( Q_f^* \) that maximises \( L \) for a given \( Q_g \). Using the calculus of variations, it is a standard result of variational Bayes that

\[
Q_f^* = \arg \max_p \frac{p(f)}{Z(Q_g)} \exp \left[ \int Q_g \log p(y|f, g) dg \right]
\]

where \( Z(Q_g) \) is the normalising constant so that \( Q_f^* \) integrates to one; that is

\[
Z(Q(g)) = \int \exp Q_g \log p(y|f, g) dg p(f) df.
\]

Inserting this into \( L(Q(f), Q(g)) \) gives

\[
F(Q(g)) = \log Z(Q(g)) - D_{KL}(Q(g)|P(g)) \quad (2.1)
\]

We can constrain \( Q \) to be a multivariate normal distribution so that \( Q(g) = N(g|\mu, \Sigma) \). Inserting this into the above equation gives:
\[ L(\mu, \Sigma) = \log N(y|0, K_f + R) - \frac{1}{4} tr(\Sigma) - D_{KL}(N(g|\mu, \Sigma), N(g|\mu_01, K_g)) \]

This depends on the \( n + n(n + 1)/2 \) parameters of \( \mu \) and \( \Sigma \). Using the fact that at the minimum the differential of \( L \) with respect to \( \mu \) and \( \Sigma \) must be 0, we find that

\[ \mu = K_g(\lambda - \frac{1}{2})I + \mu_01, \quad \Sigma^{-1} = K^{-1} + \Lambda \]

for some positive semidefinite matrix diagonal matrix \( \Lambda \). Therefore we can reparameterise equation 2.1 to depend only on \( \Lambda \) and hence it only has \( n \) free elements. This means that the log likelihood can be calculated in \( O(n^3) \) time.

The optimal parameters are calculated using maximum-likelihood estimates in the same way as with the standard GP case.

Figure 2.9 shows HGPs fitted to two intervals from different time-series of accelerometer data. It may be seen that the confidence intervals vary during each wave, as required, whereas a conventional (homoscedastic) GP would have constant posterior variance for these equi-sampled data. This is important for our application because, as described previously, the noise profile is not constant throughout each wave, and therefore the posterior variance should not be constant throughout.

The lower panels in the figure show the log of the variance \( g(t) \). We can see that the variance is as we would expect: it takes higher values on the upwards part on the curve and lower values on the downwards part. In the first of the two examples shown in the figure (leftmost), the noise function is smooth and periodic, meaning that the noise process was approximately consistent between pumping actions. In the second example
(rightmost), the noise process takes a very different shape between successive waves, meaning the vibrations vary substantially. These differences in noise profile will be seen to be important for characterising aquifer depth, when we describe the next layer in the GP hierarchy, below.

![Figure showing HGP applied to two samples of different recordings](image)

**Figure 2.9:** Figure showing HGP applied to two samples of different recordings. Top: 96Hz data; Middle: 96Hz data (blue), latent mean function (red), 95% confidence interval (grey); Bottom: latent variance function (red) with respective 95% confidence interval.

We fit an HGP to a recording $r$ to get functions representing the shape $s_r$ and vibrations $v_r$.  


Chapter 3

Modelling depth and results

We use the feature vectors for shape $s_i$ and vibration $v_i$ for wave $w_i$ to make depth predictions. We compare two regression models for depth prediction: support vector regression and Gaussian processes.

3.1 Support Vector Regression

Support vector regression (SVR) is an extension of support vector machines, a kernel-based classification technique.

Suppose we have data \{$(x_1, y_1), \ldots, (x_n, y_n)$\} where $x$ is $p$-dimensional with label $y$. To fit a linear model, we find $w$ such that $f(x) = w^T x + b$ for $w \in \mathbb{R}^p, b \in \mathbb{R}$. We want $w$ to have minimal complexity as measured by $||w||^2$. It is unlikely that a model can fit $f$ perfectly, and so we fit slack variables $\xi$ such that $f(x) = w^T x + b + \xi$, where $\xi$ is a vector of random variables with mean 0. Therefore, we want to minimise:

$$||w||^2 + C \sum |\xi_i|$$

subject to

$$y_i - w^T x_i \leq \xi_i \text{ for all } i.$$
The optimisation is solved using Lagrangian multipliers. These give:

\[ w = \sum \alpha_i x_i \]

where \( \alpha_i \) are the Lagrangian multipliers. Only some of these will be non-zero, and the corresponding \( x_i \) are known as the support vectors. Equivalently, when calcluating \( w \), only a proportion of \( x \) need to be taken into account, and this sparsity makes SVR efficient for large datasets.

SVRs can easily be extended to non-linear models using the “kernel trick”. That is, we can map \( x \) with some function \( \phi : \mathbb{R}^p \rightarrow \mathbb{R}^{p+k} \) for some \( k \geq 0 \). As \( k \) becomes larger, it quickly becomes infeasible to deal with so many dimensions, and so instead kernels are used as a means of implicit mapping. We use the squared-exponential kernel:

\[ k(x_i, x_j) = \exp(-\gamma \| x_i - x_j \|^2) \]

\( C \) and \( \gamma \) are parameters of the model to be found which correspond to radius in the \( p \)-dimensional space and error sizes. To find their optimal values we use four-fold validation with a grid search. We search over values of \( C = 2^{-5}, 2^{-3}, \ldots, 2^{13}, 2^{15} \) and \( \gamma = 2^{-15}, 2^{-13}, \ldots, 2^1, 2^3 \). For each pair of parameters \( \{C, \gamma\} \), we divide the training data into four sets at random, train the data on three sets and test on the remaining set. This is performed with each set as the hold-out test set; i.e four-fold cross-validation. This gives an average error for each pair of parameters, and we choose the pair that gives the lowest error.

27
3.2 Gaussian processes

Gaussian processes are easily extended to $n$-dimensional inputs, as they only depend on the distances between inputs rather than on the inputs themselves. Therefore we need to be able to define distances between pairs of inputs.

3.2.1 Distance metrics

We investigate two different distance metrics: absolute distance for HGP and wavelet features, and Hellinger distance for HGP features.

To find the absolute distance between two waves $x_i$ and $x_j$ with corresponding feature vectors for the shape $s_i = s^1_i...s^p_i$ and for vibrations $v_i = v^1_i...v^p_i$

\[
d_{s,i,j} = \sum_{r=1}^{p} |s^r_i - s^r_j|
\]

\[
d_{v,i,j} = \sum_{r=1}^{p} |v^r_i - v^r_j|
\]

As the feature vectors for shape and vibrations are on different scales, they must be normalised to be able to combine them into a single measure.

\[
d_{i,j} = \frac{d_{s,i,j}}{\text{mean}_{k,l=1,...,n} d_{k,l}} + \frac{d_{v,i,j}}{\text{mean}_{k,l=1,...,n} d_{k,l}}
\]

At each point of our feature vector, we have a probability distribution $z_r \sim N(s^r_i, v^r_i)$. There are many metrics used for probability distributions, and we use the Hellinger distance:
\[ D^2 = \frac{1}{2} \int \left( \sqrt{p(x)} - \sqrt{q(x)} \right)^2 \, dx = 1 - \int \sqrt{p(x)q(x)} \, dx \quad (3.1) \]

If \( p(t) \) and \( q(t) \) are Gaussian, this reduces to

\[ D_{i,j} = 1 - \sqrt{\frac{2\sigma_i \sigma_j}{\sigma_i^2 + \sigma_j^2}} \exp \left[ \frac{1}{4} \frac{(\mu_i - \mu_j)^2}{\sigma_i^2 + \sigma_j^2} \right] \quad (3.2) \]

and so we have

\[ d_{i,j} = 1 - \sum_{r=1}^{P} \sqrt{\frac{2v_r^i v_r^j}{(v_r^i)^2 + (v_r^j)^2}} \exp \left[ -\frac{1}{4} \frac{(s_r^i - s_r^j)^2}{(v_r^i)^2 + (v_r^j)^2} \right] \]

The distances are presented to Gaussian processes with the following squared-exponential covariance function, giving a positive semi-definite kernel:

\[ K(x_i, x_j) = \sigma_j^2 \exp \left( -\frac{D_{i,j}^2}{2l^2} \right) \]

### 3.3 Results

#### 3.3.1 Testing process

We evaluate our various methods via two cross-validation schemes. In the first scheme, denoted “randomly held-out waves”, we hold out 25% of individual waves selected at random as a validation set within each fold of cross-validation, and use the remainder as a training set. In the second scheme, “randomly held-out recordings”, we take 30% of recordings and hold out all of their constituent waves as a validation set within each...
fold of cross-validation, and train on the waves from the remaining 70% of recordings. Holding out entire recordings is a more difficult test than holding out waves at random, but is arguably more fair - otherwise waves from a single recording may appear in both training and validation sets, which represents a potentially unfair performance advantage that would not be seen in practice.

We observe that a large proportion of our data is concentrated around the average aquifer depth. To determine whether or not this biases our results, we use an additional third scheme, in which the training set is balanced. We divide all recordings into three quantiles, and take an equal number of waves from each of the shallow, middle, and deep quantiles. Finally, this balanced set is further subdivided at random into a training and validation sets of equal size.

In all three schemes, the parameters of the various models in the hierarchy are determined by minimising the NLML on the training set, before being used to estimate the depth of the aquifer for the corresponding validation set.

Each individual wave \( w_i \) results in a depth prediction \( y^* \) for that wave and so we obtain multiple predictions \( y^*_i \) for each recording. When evaluating error in our depth prediction, we consider the cases (i) when we use each individual \( y^* \) on a wave-by-wave basis (effectively treating each wave in a recording as being independent), and (ii) when we define a single depth prediction for a recording by taking the median of the \( y^* \) for that recording, which we denote \( y^*_m \).

We test all the different feature vectors, and, where applicable, the different distance metrics. We use feature vectors of length \( p = 8, 16, 32, 64, 100 \) from sections 2.2.1 - 2.2.3. Note that in the case of smoothing splines, the cases of \( p = 64, 100 \) due to the number
of data in a wave \( w_i \). Having fitted the model and calculated the errors for each \( p \), we choose the optimal value of \( p \) as being that with the lowest cross-validation error. For depth prediction, each of our two methods, GPR and SVR have advantages. In a general regression problem, SVRs often outperform other regression methods, as they are specifically optimised to minimise the prediction errors. GPR, on the other hand, give estimates of uncertainty along with point estimates, which will be of use when using the point-depth predictions. With the HGP feature vector, we use Gaussian process prediction for a consistent probabilistic approach. We chose to use SVR with smoothing splines and GPR with wavelet features.

### 3.3.2 Kellogg dataset

Table 3.1 shows the median absolute errors, with the interquartile range, in each case. Figure 3.1 show violin plots for each case, showing the spread of individual wave predictions for each recording, along with their median.

We noted above that holding out by recording is a much harder test than holding waves out at random, and indeed we see that we have higher errors when holding out by recording. In general, the errors of \( y_m^* \) are lower than the errors of \( y^* \). This is expected, as we would expect the individual predictions for a recording to be centred around the true value, so that when we take their median it is more accurate than the individual \( y^* \). However, there are some cases, where the error for \( y^* \) is lower than that for \( y_m^* \). This happens when many recordings have very accurate predictions, but some recordings have extremely inaccurate ones with a large bias, so that the median is also very inaccurate.

Considering the HGP-based feature vectors, we see that which features give the best
Table 3.1: Randomly held-out waves

<table>
<thead>
<tr>
<th></th>
<th>Individual</th>
<th>Recording</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shape</td>
<td>6.09 (2.69, 12.56)</td>
<td>6.50 (2.29, 13.00)</td>
</tr>
<tr>
<td>HGP &amp; GPR</td>
<td>6.61 (2.81, 15.38)</td>
<td>4.47 (1.95, 17.05)</td>
</tr>
<tr>
<td>Combined</td>
<td>5.13 (2.51, 10.59)</td>
<td>4.06 (1.67, 7.65)</td>
</tr>
<tr>
<td>Hellinger</td>
<td>5.25 (2.2, 11.44)</td>
<td>4.76 (1.66, 9.94)</td>
</tr>
<tr>
<td>Splines &amp; SVR</td>
<td>7.11 (2.93, 17.19)</td>
<td>5.55 (2.33, 16.22)</td>
</tr>
<tr>
<td>Shape</td>
<td>6.50 (2.29, 13.00)</td>
<td></td>
</tr>
<tr>
<td>HGP</td>
<td>4.8 (2.25, 10.29)</td>
<td></td>
</tr>
<tr>
<td>Combined</td>
<td><strong>3.67</strong> (1.57, 7.88)</td>
<td><strong>2.68</strong> (1.01, 6.38)</td>
</tr>
<tr>
<td>Wavelets &amp; GPR</td>
<td>5.61 (2.54, 11.11)</td>
<td>4.62 (1.48, 10.05)</td>
</tr>
<tr>
<td>Vibration</td>
<td>9.52 (4.19, 18.87)</td>
<td>7.22 (3.11, 19.35)</td>
</tr>
<tr>
<td>Combined</td>
<td>5.5 (2.39, 10.8)</td>
<td>3.63 (1.67, 8.91)</td>
</tr>
</tbody>
</table>

Table 3.2: Randomly held-out recordings

<table>
<thead>
<tr>
<th></th>
<th>Individual</th>
<th>Recording</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shape</td>
<td>7.01 (3.50, 13.38)</td>
<td>7.25 (3.37, 11.32)</td>
</tr>
<tr>
<td>HGP</td>
<td>8.38 (3.35, 19.87)</td>
<td>5.06 (2.54, 19.87)</td>
</tr>
<tr>
<td>Combined</td>
<td>6.82 (3.13, 14.36)</td>
<td>5.75 (2.42, 12.55)</td>
</tr>
<tr>
<td>Hellinger</td>
<td>7 (3.17, 14)</td>
<td>5.78 (2.47, 13.41)</td>
</tr>
<tr>
<td>Shape</td>
<td>8.88 (3.19, 26.72)</td>
<td>5.81 (0.68, 26.84)</td>
</tr>
<tr>
<td>SVR</td>
<td>6.73 (3.19, 14.57)</td>
<td>4.14 (2.01, 16.59)</td>
</tr>
<tr>
<td>Combined</td>
<td><strong>5.51</strong> (2.37, 13.48)</td>
<td><strong>2.94</strong> (1.05, 11.61)</td>
</tr>
<tr>
<td>Wavelets</td>
<td>7.64 (3.43, 16.7)</td>
<td>6.69 (1.76, 15.73)</td>
</tr>
<tr>
<td>Vibration</td>
<td>10.59 (4.67, 21.94)</td>
<td>7.99 (2.37, 21.37)</td>
</tr>
<tr>
<td>Combined</td>
<td>7.92 (3.51, 15.22)</td>
<td>5.92 (2.44, 15.81)</td>
</tr>
</tbody>
</table>

Table 3.3: Balanced data (randomly held-out waves)

<table>
<thead>
<tr>
<th></th>
<th>Individual</th>
<th>Recording</th>
</tr>
</thead>
<tbody>
<tr>
<td>HGP/GPR</td>
<td>4.74 (2.07, 9.65)</td>
<td>2.42 (1.14, 5.12)</td>
</tr>
<tr>
<td>Splines/SVR</td>
<td>3.16 (1.26, 7.85)</td>
<td>2.11 (0.87, 6.4)</td>
</tr>
<tr>
<td>Wavelets/GPR</td>
<td>6.64 (3.01, 11.81)</td>
<td>3.9 (1.24, 8.06)</td>
</tr>
</tbody>
</table>

Table showing median (IQR) errors in cm for each of our feature vector/prediction models, with each method of testing. The first column refers to model used. The second column refers to feature vector used; we use each of the “shape” and “vibrations” feature vector separately, and then the two combined, as described earlier in this section. In the case of HGP, we have two ways of combining them: “Combined” refers to using the absolute distance metric, and “Hellinger” refers to using Hellinger distance metric. On the balanced dataset, we used the “combined” feature vector.

Table 3.4: Kenya data (randomly held-out waves)

<table>
<thead>
<tr>
<th></th>
<th>Individual</th>
<th>Recording</th>
</tr>
</thead>
<tbody>
<tr>
<td>HGP/GPR</td>
<td>1.60 (3.77, 7.11)</td>
<td>1.88 (4.05, 6.74)</td>
</tr>
</tbody>
</table>

Table showing median (IQR) errors in m for the HGP/GPR model on the Kenya dataset.
Figure 3.1: Violin plots showing results on the randomly selected test set for all three of our methods. Along the x-axis is each recording, and the ‘violin’ shows the spread of individual predictions from the waves of that recording.
results depends on the testing method. As an example, the errors of $y^*_m$ when holding out recordings are lowest when using the vibration vector (5.06cm), slightly higher using the combined features for either distance metrics (5.75cm, 5.78cm) and highest when using the shape vector $s$ (7.25cm). Interestingly, the errors of $y^*$ are lower when using the shape rather than vibration vectors, yet the errors of $y^*_m$ are lower when using the vibration vectors. This suggests that the shape vector $s$ gives good results for most recordings, but on some recordings it has a large bias, whereas the vibration vector $v$ gives good results more consistently over the recordings.

Figure 3.1 shows a violin plot of the spread of the results for each recording with their median. The accuracy of the results depends on the true depth. For those of medium depth (recordings 25 - 95), the $y^*$ for each recording is centred around the true depth with a small spread on either side. Therefore, the errors for $y^*$ will be low for these predictions, and the errors for $y^*_m$ even lower. For our data with an increased depth (recordings 96 - 116) we can see again that we have predictions on either side of the true depth. However, there is a large spread in the predictions for the individual waves of these recordings, which increases the errors of $y^*$. Moreover, there is a bias in the predictions, such that many more predictions are below the true depth, which means that the error of $y^*_m$ will also be high. For the shallowest recordings (1 - 24), this problem is amplified, but less consistent. Many of the recordings have a very large spread, from far shallower to the range of the true depth to far deeper. Moreover, there is a bias such that in all but one of the recordings $y^*_m$ is higher than the true depth. In some cases, where the spread is not so large the true depth does not even come in the range of $y^*$.

When using the smoothing spline feature vector and SVR predictions, the vibration
vector $v$ gives lower errors than the shape vector $s$, of 4.14cm on $y^*_m$ compared to 5.81 cm, meaning there is more information relating to depth in the vibrations of the handle rather than in the shape of the pumper’s motion, which we might expect. Combining $s$ and $v$ gives the best predictions with errors of 5.51cm for $y^*$ and 2.94cm for $y^*_m$. We note that they are the best predictions across all features. This may be because, all other things being equal, SVRs will normally give a better prediction than Gaussian processes, as they are specifically designed to reduce the residual error, in a way that Gaussian processes are not. However, looking at figure 3.1 we see a similar pattern of errors as we do in the HGP/GPR case. That is, the predictions for recordings 25 - 95 have low bias and spread, but recordings 96 - 116 a larger spread and a negative bias, and recordings 1 - 24 have an even larger spread and a positive bias.

In contrast to the spline/SVR case, when using wavelets the shape $s$ gives much better results than $v$. Indeed, for the most part, the wavelets $v$ gives worse results than other feature matrices and vectors on each test, e.g. 7.99cm for $y^*_m$ when holding out recordings. It is likely that the high-frequency wavelets fail to contain useful information unlike the other vibration representations. Looking at the example in figure 2.8, we can see that we do not get a distinctive periodic pattern in the wavelet vibrations as we do in the HGP or SVR vibrations, and this indicates that there is no distinguishing information in each wave. However, using shape and vibrations combined, we do get reasonable results; e.g. 5.92cm on the same test. However, looking at figure 3.1, we notice that again recordings 1 - 24 have a large spread and positive bias while recordings 25 - 95 have low bias and spread and recordings 96 - 116 have a negative bias.

We fitted the models to a dataset that is balanced by selecting an equal number of
waves from the depth categories \( \{ < 4.1 \text{m}, 4.1 < 4.4, > 4.4 \text{m} \} \). In table 3.4 we see that in fact we get lower errors on the balanced data set than on the full one, e.g. median error of 2.11cm on \( y_m^* \) using splines/SVR compared to 2.94 on the full dataset.

### 3.3.3 Kenya dataset

We also fitted the HGP/GP model to the Kwale dataset, by way of example. The results are shown in figure 3.2. We see that the predictions are reasonable for the shallowest and deepest pumps, from which most of our data comes. However, once again there is a bias in the predictions such that the shallowest depths (recordings 1 - 37) are overestimated and the deepest ones (recordings 59-85) are underestimated. The model also fails to discriminate well between those with medium depths (38-58) possibly because we have very little data for each point in that range.

![Figure 3.2: Violin plots showing violin plot of results of the HGP/GPR model on the Kenya dataset.](image-url)
Chapter 4

Relationship between groundwater depth and tide height

We describe a further analysis using data from a handpump at Jabalini in Kwale, which is located around 100m from sea. There is no great rise in altitude from the sea to the location of the pump, and the well is approximately 10m deep. Since April 2014 a depth logger has been put in the well. Figure 4.1 shows this depth, or groundwater level, between April and September. We can see that there are interesting periodical patterns. Also shown is the tide height at Mombassa, approximately 60km up the coast from Jabalini.

Tides refer to the phenomenon of the sea level at one location varying due to the effect of the gravity of the moon and the sun. Most places on the earth experience two high tides (and low tides) per day, normally of different heights. When the sun and moon are aligned, the tidal range is larger, known as springs, and when they are perpendicular to each other, the tidal range is smaller, known as neaps.

In figure 4.1 we can clearly see these springs and neaps, seperated by intervals of a week. Figure 4.2 shows the same over approximately four weeks, and we can see the two tides per day with different heights. The tide height and groundwater level are
normalised so that each has mean zero and unit standard deviation. Figure 4.1 also shows the groundwater level over the respective periods. Each day, its depth tracks tide height with a slight lag, meaning that as the tide rises water spills into the groundwater increasing its height. Interestingly, the average groundwater level is higher during neaps and lower during springs. This is surprising, as although the tidal range changes, the mean depth of water stays the same.

![Figure 4.1: Plot of tide height and groundwater depth over a period of five months. Note that depth is negative so that they are both shown relative to groundlevel.](image)

We want to be able to model groundwater level given tide height. There are three main factors affecting groundwater depth that work over different time scales. Over a matter of hours, the depth is being affected by the daily tide variation. Over timescales of days, there is a major effect caused by springs and neaps. Over timescales of months, the level will change due to rainfall and hydrogeological effects. We can see that in the first month, the average groundwater is shallower than in later months. Clearly this longer effect cannot be modelled using tide information.
We start with modelling the daily effect. To remove longer-term effects, we create a new vector of groundwater level relative to the level at the previous low tide. The period from one low tide to the next is approximately half a day. The tide data is measured at one minute intervals and groundwater depth at 15 minute intervals. We take 50 equally spaced samples of measurements between each period from one low tide to the next. Figure 4.3 shows the new time series.

Linear regression

We model relative depth using linear regression: $Y = a^T X + \epsilon$ where $Y$ is the $(1 \times n)$ vector of relative depth, $X$ is the $(p \times n)$ matrix of inputs, $a$ is a $(p \times 1)$ vector of coefficients and $\epsilon$ is a vector of normal random variables. The inputs for $X$ are:

1. $\sin \left( \frac{t - a}{l} \right)$, where $t$ is time since low tide and $a$ is a constant to offset the time between low tide and lowest groundwater level and $l$ is to scale the frequency.

2. Tide height at previous low tide
3. Difference in height at previous low tide and the value previous to that

4. Whether changing from spring to neap or from neap to spring (binary variable)

5. Whether the tide is flooding or ebbing (binary variable)

Note that we do not use the tide height itself, but use $\sin(t-a)$ instead. This is because, given the height of the previous two low tides, the height over the period is almost deterministic, and has a sinusoidal shape, with a minimum occurring shortly after low tide. As we are using a linear model, we must therefore use $\sin(t-a)$ rather than $t$. We fitted the model using all linear and quadratic terms. Figure 4.3 shows the fitted values, and we can see that the model fits very well. This shows that a large part of the daily movement in groundwater level is determined by some characteristics of the tide height in a small window around the day.
Chapter 5

Future work

5.1 Depth prediction

By creating a prediction for each wave, we generating several estimates for groundwater level. Currently, to generate our secondary prediction, $y^*_m$, we take the median. However, we should be able to improve this by generating our secondary prediction more intelligently; for example by taking the error of the individual prediction into account.

Using the Kellogg dataset, we have been predicting the exact groundwater level. However, for the wider project, we may decide that predicting the exact level at all handpumps is infeasible, and that at some we will predict proxies instead; for example, whether the level is increasing or decreasing.

5.2 Data Collection

The immediate priority for future work is the collection of gold-standard data from Kwale. We have developed high-frequency accelerometry loggers for this purpose, based on the prototypes used for the lower-frequency water point devices. They consist of a circuit board with a high frequency accelerometer and an off-the-shelf data logger. The accel-
erometer outputs measurements similar to the Wii-mote, and these are recorded to a micro-SD card by the data logger. In order to collect the true depths, we will also fit the wells with depth loggers.

We aim to have five of these prototype accelerometry loggers, and hence we will need to choose five handpumps from which to collect data. I have identified features that distinguish pumps that will need to be taken into account when choosing which pumps we will fit the loggers to.

Of these, the most important feature is the altitude at which the pump sits, which affects the baseline depth of the well. It is important that our method works across a large range of depths, so that it can be fully transferable between different handpumps across a region. Handles of pumps are intentionally heavy to counteract the weight of the water being lifted. The handles are also adjustable so that they can be lengthened for deeper pumps to provide more leverage. However, for a deep pump, the weight and length of the handles do not fully compensate for the weight of the water, and so it still requires much effort to lift the water, and very little effort to reset the pump. Conversely, on a shallow pump the weight of the handle over-compensates for the weight of the water being lifted, so that lifting the water requires very little effort, but lifting the handle to reset the pump may require more effort. These factors change the shape of the pumping motion and most likely the vibrations through the pump.

The characteristics of pumps, both between different pumps and over time. The joint of the top of the handle and the top of the pump can rub causing different patterns of vibrations. We would expect the rubbing would lessen as the pump is used and the metal wears. There are also parts inside the rising main that are subject to substantial friction,
causing wear and tear and hence these parts have a short life cycle. Rust can also build up inside the rising main, increasing the amount of friction. The algorithms that we develop will need to be robust to difference due to the state of the pump.

When choosing pumps from which to acquire preliminary data, we will need to make sure we survey pumps exhibiting a range of these features. We will define three categories of altitude and select one or two pumps from each. We could select pump one in each altitude category to have similar degrees of smoothness as each other, and then others that are particularly smooth or stiff to acquire data from a range of conditions.

The Oxford team plans to go to Kenya to select handpumps for use, as described above and install the devices. Devices could be left for a period according to their expected battery length, with data uploaded by local partners.

This dataset will be used to design algorithms based on methods already explored. We will start by repeating our analyses of the Kellogg dataset. This will give further insight of the robustness and efficacy of our algorithms be, as we will have a greater quantity of data from an in-service setting. We will also be able to compare the data from different pumps, to see how to make the algorithms applicable to different pumps.

5.3 Large scale deployment

The long-term plan for a depth-tracking system of validated robustness is to deploy it, replacing the current handle-tracking system. The new systems will transmit the data to a central server via the GPRS network, which would provide information on groundwater depth. During our initial work, we have recorded the accelerometry data at 96Hz. However, it is infeasible to transmit that high a quantity of data, and so instead we will
transmit summaries of the data, after preprocessing on the pump. The summaries will then be processed on the central server to perform aquifer tracking over wide areas.

A key consideration is battery power within the embedded systems. The current transmitters are expected to last two years between battery changes, but the depth tracking system will almost certainly require a larger amount of battery power. Further measures can be taken to reduce power demand intelligently. For example, fifteen minutes of pumping produces around 900 waves from which predictions of aquifer depth can be made using the techniques described in this work. It may well be that we only need to transmit a short interval of data rather than data for an entire day. This could be optimised automatically. For example, 15 minutes of data could be sampled and an estimate of aquifer depth made, along with an estimate for the error of that prediction. If the error is small enough, then we may decide no further data need be transmitted for some time, e.g., a day. If the estimated error is large, however, then more data can be collected, until a suitably small estimate of the error is reached.

5.4 Groundwater depth and tides

The relation between tidal height and groundwater levels can also be modelled using multitask Gaussian processes [33]. These allow both time series to be modelled using GPs that can be linked.

Multitask GPs at their most general can model a number of processes, sampled at different rates or over different timescales. In our case, we use two tasks, sampled at the same times, tide height $h$ and groundwater depth $d$. The MTGP has a covariance matrix giving the covariance between time $t_i$ and $t'_j$ for tasks $i, j = \{h, d\}$. We assume
that the covariance functions between tasks and between different times within a task are independent of each other, such that: \( k(t, t', h, d) = k_c(h, d) \times k_t(t, t') \) where \( k_c \) is the covariance matrix for the tasks and \( k_t \) is the covariance matrix between times, similar to the univariate case. We use this to create a covariance matrix, again using the squared-exponential function. We introduce a further parameter \( s \) to allow for the time lag between the two series.

We train an example MTGP using data from days 1 - 49 to estimate values of the hyperparameters, using NLML as before. Figure 5.1 shows the resulting MTGP applied to days 50 - 80. We see that the fit is worse than that of linear regression, shown before in fig 4.3. This is because the prediction for groundwater depth at time \( t' \) is using only that information concerning the tide height from a short window around \( t' - s \), whereas our linear regression uses information from a wider time-window. However, we note that the troughs and peaks of the true and fitted data line up well, implying that the time-shift parameter has been accurately learnt. We will incorporate information from a wider time window by creating Gaussian processes for the other features, including the height of the highest daily tide, and the difference between the two daily high tides, which can further be linked into the MTGP model.

5.5 Wide-area estimation of groundwater

We may obtain point estimates of aquifer depth at handpumps locations as described in this report which we will use to find a smooth estimate of groundwater level across a wide area. This a long-standing problem in geostatistics and many methods have used. We will experiment with different models for the collated prediction. We will also have
error estimates for the depth at each point, which can be incorporated into the collation method to provide error estimate for the level across the area as well. The models and estimates will also be informed by hydrogeology studies that are taking place within the project.

We can also use the collation to improve the intelligent sampling of data and battery conservation. For example, if we have three pumps very close together, we may sample from just two of them, only sampling from the third if first the two give inconsistent results.

5.6 Condition monitoring

A further aim of the project is to be able to monitor the condition of the pumps and be able to predict if one is about to breakdown. When collecting data for condition
monitoring, there is generally far more data from times of “normal operation” than from failures. For this reason, condition monitoring generally relies on forms of one class classification. That is, the “normal” behaviour is learnt, so that anything considered to be outside of normal behaviour can be identified as a sign of possible failure. During the prototype stage, it is unlikely that a pump will fail and so we will only have data from normal operation; we can thus learn the normal behaviour of handpumps during this stage. At the large-scale deployment stage, however, we will likely have much data from minor failures, such as the degradation of rubber parts in the rising main, and probably some examples of major failures which will allow us to develop methods for identifying specific failure modes.

5.7 Timeline

<table>
<thead>
<tr>
<th>Data collection</th>
<th>MT15</th>
<th>HT16</th>
<th>TT16</th>
<th>LV16</th>
<th>MT16</th>
<th>HT17</th>
<th>TT17</th>
<th>LV17</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gold standard data collection</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Large scale roll out</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Analysis</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Depth prediction on gold standard data</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Condition monitoring - theory, model using g’s data, incorporate field data</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tides and groundwater depth</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wide-area prediction</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Analysis of large scale data</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Writing thesis</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Bibliography


