INTERNATIONAL JOURNAL OF
NUMERICAL ANALYSIS AND MODELING
Volume 9, Number 2, Pages 169–180

REAL-TIME MULTISCALE DETECTION OF DEFECTIVE PILLS DURING MANUFACTURING

CRAIG C. DOUGLAS, LI DENG, YALCHIN EFENDIEV, GUNFOLF HAASE, ANFREAD KUCHER, AND ROBERT A. LODDER

(Communicated by Zhangxin Chen)

Abstract. We explore methods to automatically detect the quality in individual or batches of pharmaceutical products as they are manufactures. The goal is to detect 100% of the defects, not just statistically sample a small percentage of the products and draw conclusions that may not be 100% accurate. Removing all of the defective products, or halting production in extreme cases, will reduce costs and eliminate embarrassing and expensive recalls. We use the knowledge that experts have accumulated over many years, dynamic data derived from networks of smart sensors using both audio and chemical spectral signatures, multiple scales to look at individual products and larger quantities of products, and finally adaptive models and algorithms.

Key words. manufacturing defect detection, dynamic data-driven application systems, DDDAS and integrated sensing and processing, high performance computing, and parallel algorithms

1. Introduction

Diabetes is a problem worldwide. Of the more than 15 million Americans who have diabetes mellitus, about five million do not know it. Nearly 1 million new cases are diagnosed each year. The disease affects men and women of all ages and certain ethnic groups are more greatly affected than other groups [1]. With the more common type 2 diabetes, the body does not make or use insulin properly. Without enough insulin, the glucose stays in the blood system. Having too much glucose in the blood system causes serious problems, e.g., damage to the eyes, kidneys, and nerves. Other side effects of diabetes include heart disease, stroke, and removal of limbs. Pregnant women can also have gestational diabetes [2].

The total annual economic cost of diabetes in 2007 in the U.S. was estimated to be 32% higher than just five years earlier. In 2007, medical expenditures totaled \$116 billion (\$27 billion for diabetes care, \$58 billion for chronic diabetes-related complications, and \$31 billion for excess general medical costs). Indirect costs resulting from increased absenteeism, reduced productivity, disease-related unemployment disability, and loss of productive capacity due to early mortality equaled the cost of chronic diabetes-related expenditures. One out of every five health care dollars is spent caring for someone with diagnosed diabetes, while one in ten health care dollars is attributed to diabetes [3].

While U.S. drug products are of generally high quality, there is an increasing trend toward manufacturing-related problems that lead to recalls, disruption of manufacturing operations, and loss of availability of essential drugs. The U.S. Food and Drug Administration (FDA) is combating these problems plus low manufacturing process efficiency (<30%) has also led to increased cost of drugs by emphasizing current good manufacturing practice (cGMP) as the means of controlling drug quality. An unfortunate side effect is that many companies are no

Received by the editors November 20, 2009 and, in revised form, April 2, 2010.
2000 *Mathematics Subject Classification.* 62H35, 68U10.

longer innovating at the same rate as before. The FDAs response is that uses of new sensing technologies will be key to improving the regulation and quality of drug manufacturing using scientifically proven risk-based methods. Near infrared (NIR) is one of the process analytic technologies (PAT) that the FDA has chosen to improve manufacturing process quality [4].

In Section 2, we describe the smart sensors we are designing for use on manufacturing lines and how both offline high performance computing and integrated sensing and processing are involved.

In Section 3, we describe where multiscale techniques are useful and how to construct them.

In Section 4, we describe the process of finding the best frequencies for drug identification considering algorithmical and numerical aspects.

In Section 5, we describe some identification results along with some simple timing information for the parallel computation needed to create libraries for our sensing devices.

In Section 6, we draw some conclusions.

2. Smart Sensors with Integrated Sensing and Processing

Smart sensors are a form of integrated sensing and processing (ISP). ISP optimizes systems that integrate sensing, signal processing, communication, and targeting. Traditional sensing systems lead to high dimensional and prohibitively expensive problems to solve. ISP based methods lead to reduced and low dimensional systems that can be solved through a combination of the on board computing on the sensors and by solving auxiliary problems on highly parallel computers in advance. We convert data directly to knowledge using programmable on demand ISP-based imaging spectrometers that produce detector signals that can be correlated directly to desired samples. Hence, we do not need to do a post collection chemometrics step. Parallel computers produce libraries that are downloaded to the ISPs and include environmentally relevant information that is updated on a regular basis. While these libraries are expensive to create computationally, the use in the ISPs is both inexpensive and fast.

Dynamic data-driven application systems (DDDAS), [5], [6], and [7], place far different strains on high performance systems and centers than traditional applications due to dynamic and unpredictable changes in resources that are required during long term runs. An on demand environment is required that stresses traditional computing centers views on allocating resources. The number of processors, network resources, and location of computing and data can change unpredictably during the course of a long term DDDAS computation [8] and [9].

DDDAS assumes that application components, resource requirements, application mapping, interfaces and control of the measurement system can be modified during the course of the application simulation. Figure 1 provides a schematic of how typical DDDAS applications appear componentwise.

The longest running DDDAS we are aware of began its calculations in 1978 and still runs today even though all of the hardware and software have changed over time, but the application has never been turned off. The application monitors all oil and gas pipelines, storage tanks, wells, and tanker loading in Saudi Arabia.

The devices we are designing for this project are designed to reduce the over abundance of data that is prevalent in most pharmaceutical manufacturing environments today [10]. We envision using similar devices in related fields, e.g., handheld devices to ensure correct pill delivery wherever health caregivers are involved.

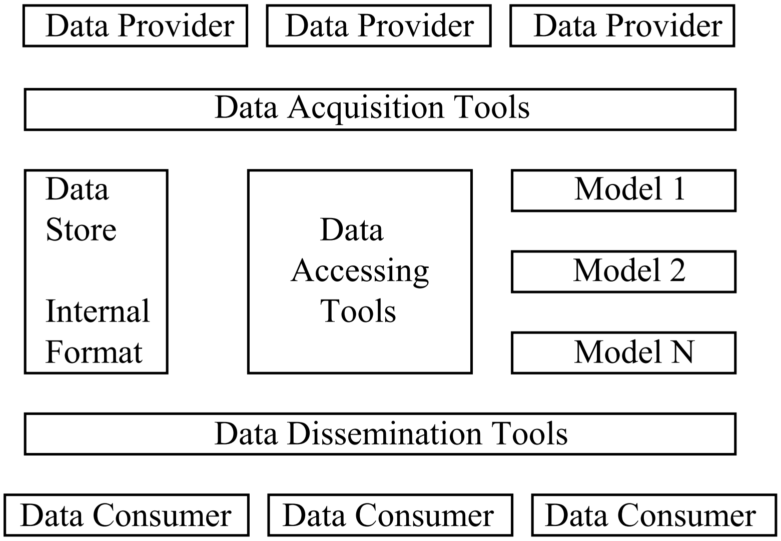

Figure 1. A typical DDDAS

Providing a reprogrammable, networked embedded system that can automatically determine quality and control end products in a manufacturing line will impact not only the pharmaceutical industry, but any similar chemically-based production facility. A likely offshoot is a handheld acoustic device that medical providers can use to correctly identify all medications that will be given to patients. Taking the wrong medications is the fifth leading cause of death in the United States.

The identification of pills by using acoustic waves is a challenging problem in technical pharmacy. Manufacturing lines are extremely noisy and we typically only have 8-10 milliseconds to identify each defective or incorrect pill. Our goal is to catch 100% of the possible problem pills and get them out of the manufacturing line before they reach packaging.

In the U.S., a pilot line is created, tested, and eventually approved by the FDA. Once approved, it becomes a production line and cannot be modified in any significant way. Any significant modification to a production line makes it a pilot line again with all of the testing and approvals required again. Further, the FDA has to approve to conversion back to a pilot line, which it may not do.

Spherix is a small pharmaceutical company that created Tagatose [11], [12], and [13]. The manufacturing line belongs to a much larger company that is in Italy that does contract work for others. We are in a position to create a complete DDDAS and test it on a real pharmaceutical manufacturing line in Italy, where the rules for testing new pharmaceutical technology is quite different than in the U.S.. In particular, we do not have to get FDA approval to make an adjustment to the manufacturing line in order to improve the quality control mechanisms.

The system we are creating corresponds to a true DDDAS since it involves combining pill manufacturing and environmental factors with integrated sensing and processing. A number of human factors also are included based on who is working on the manufacturing line during a work shift and the peculiarities of the individual workers. Incoming raw materials vary, too, and the system needs small, but significant adjustments on a regular basis, typically timed for work shifts. The updates require some substantial high performance computing to create downloadable libraries for the actual devices.

The DDDAS is built using a cumulative set of components:

- The production process is modeled using TagSim, a semi-empirical Simulink [14] program written in Matlab. TagSim predicts both the output yield and impurities for the tagatose production process.
- Remote access to the system uses a secure client-server approach on the Internet.
- Servers are connected to the processes through standard data acquisition hardware and the Matlab Communication Toolbox. The combination provides security, data validation, and session management.

The remote access provides a secondary side effect, namely a nice mechanism for training students in process control.

The choice of Matlab speeded up the development of the entire system. In part because Matlab is well known by students, but also that it was easy to design, test, and verify control strategies in real time over the Internet. Both interlocks and cutoffs were easy to implement, both of which guarantee safe and secure operation of the remote processing units.

One problem that came to light was that computation and networking speeds have to be carefully monitored. Using remote supercomputers to generate the libraries for the ISP devices must minimize data transfers in order to be useful. It is not yet practical to transfer the data to a remote supercomputer, do a fast calculation on it, and transmit the result for an individual pill back to the manufacturing line. It is essential to do the identification on the ISP directly.

The actual acoustic devices use a combination of integrated sensing and processing and acoustic resonance spectroscopy (ISP-ARS). This is a novel approach to acoustic spectroscopy that can be implemented using instruments as simple as a MP3 player like an Apple iPod Touch$^{\text{TM}}$ or one that is extremely complex. The choice depends on the environment that the ISP-ARS will be used in. A wireless networked MP3 player is ideal since the library can be downloaded from a server at selected times or pushed by the server.

ISP-ARS is both fast and non-destructive. Unlike near-infrared or optical methods, acoustic methods are can penetrate deeply many common forms of opaque packaging. The penetration ability is a significant advantage in preparing clinical trial lots since drugs and placebos should be indistinguishable by the patients.

Pharmaceutical manufacturing lines for pills are extremely noisy environments. Simple acoustic devices are worthless unless a sound box is created to house the acoustic device while letting pills pass by. In addition, the pills move on manufacturing line at a considerable speed. At any physical region of the line, the pills are only available for defect identification for 5-10 milliseconds and must then be eliminated from the line. Further, our goal is to eliminate every single defective pill, not just some of them.

The ISP-ARS device must perform an optimization of a functional that is equivalent to solving a complicated nonlinear equation. We use automatic differentiation techniques to simplify the entire process to something that can be put into an embedded system. Numerical approximations are used to further simplify the computing tasks to something practical for the embedded system. All we care about is that the ISP computation is accurate and fast enough to catch each defective pill.

All pills sold in the world are supposed to be unique in shape, size, and coloring. All pills have a unique audio spectra footprint. We play specific noise patterns at pills and measure the resulting spectra. We look for specific spectra to identify specific types of pills. We have to identify which spectra to look for given environmental factors that may change over time as short as a work shift.

In ISP-ARS, an acoustic waveform is created that comprises just the distinguishing spectral details associated with an analyte of interest. Fourier transform acoustic resonance spectroscopy (FTARS) is used to develop ISP acoustic waveforms employed in differentiating different drugs.

As a PAT, a series of ISP-ARS sensors must scan every pill produced by a manufacturer, enabling the removal of only those pills that do not meet quality standards or controls. Measurements from a series of ISP-ARS sensors should control the manufacturing line dynamically and adjust the process conditions and ingredients in real time based on actual process measurements [15] and [16].

ISP-ARS reduces the time required for processing that normally occurs with full spectrum FTARS. An ISP acoustic waveform results from chemometric analysis of the FTARS spectrum. By weighting the frequency changes according to individual component scores, an acoustic waveform can be constructed that excites just the frequencies important to the analyte in question. The ISP output is a voltage that can be read immediately and corresponds to just the analyte in question.

The ISP acoustic wave design starts with the chemometric analysis of the initial FTARS data. FTARS makes a prediction about what works as an ISP acoustic waveform for a given set of samples. The training process can be interpreted as a DDDAS that continuously monitors the performance of the ISP waveform that is continuously adjusted through retraining. In essence, a symbiotic relation between the computation and the sensing that results in highly quality manufacturing.

Consider Figure 2. The traditional FTARS cycle is steps ABCDA. In traditional FTARS, samples are scanned and classified according to their inter-cluster distances found via multivariate analysis (steps A-D). For each sample or groups scanned the process is repeated. The FTARS data is used as a predictor for ISP-ARS. The prediction (E) is used to construct an ISP acoustic waveform. The traditional FTARS cycle is no longer needed once the ISP waveform is constructed. Samples scanned with the ISP waveform are classified according to their voltages (F-G). If a sample cannot be classified (H), then FTARS is employed to construct chemometrically a new ISP acoustic waveform usingth a training set that it includes the new unknown. As samples change during time, the ISP waveform adapts to the new data.

FTARS can be used to differentiate liquids, powders, drugs, and predict dissolution rates in seemingly identical samples. FTARS is nondestructive and complete scans can be made in seconds, therefore it should be a prime candidate for use as a PAT. Following data collection, FTARS relies on intensive computer processing due to the amount of information gained in each scan. An ARS spectrum recorded over the interval of 20 Hz to 20 kHz with a sample rate of 44.1 kHz for one second generates a large amounts of data from the 44100 data points. Chemometric analysis of multiple FTARS data sets requires too much computation, which limits the production rate of tablets, which is unacceptable. ISP-ARS is fast enough to not limit production rates since it directly produces the analyte identity as an output, thus eliminating the computational time of FTARS.

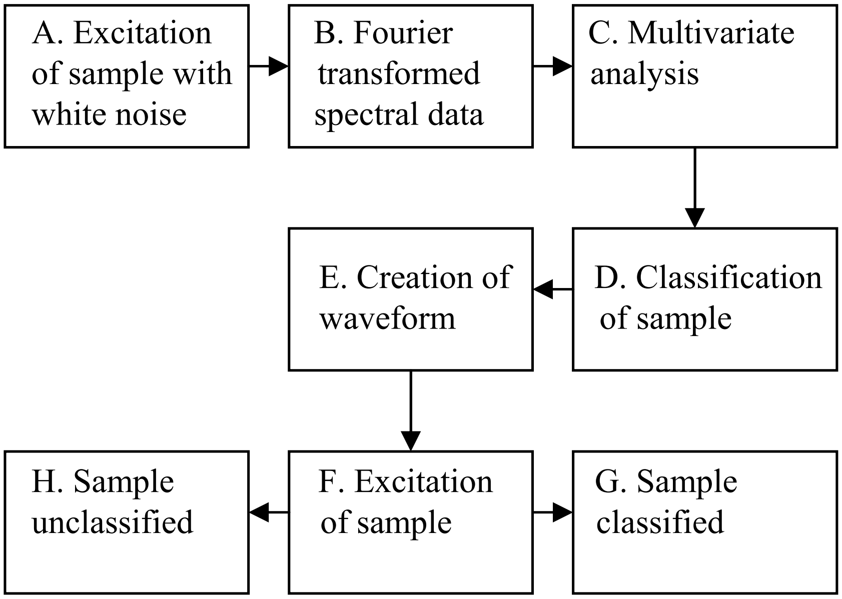

FIGURE 2. A classification of the steps in FTARS and ISP-ARS

3. Multiscale Advantages

We can assume that the ISP-ARS devices are in multiple locations in the production line and that there are local computing resources. The ISP-ARS devices are identifying defective or low quality pills on the manufacturing line as defective. Data can be transmitted to a local computer that maps the quality of the overall production line.

The map can easily be on different scales ranging from the entire production line to space corresponding to multiple pills. The advantage of using a multiscale technique is that when problems arise on the production line, we can determine quickly and automatically whether we have a minor or major problem. If enough defective pills are identified at once or in enough different locations, we may want to shut down the production line until the problems can be resolved. Shutting down the production line is both expensive and under normal circumstances not done. Doing so automatically with the problem area already identified will reduce the expenses considerably.

In packaging, identifying that possibly millions of pills are mislabeled will save a costly product recall. For a drug test, it will possibly save the entire set of human trials from having to be discarded and a new country targeted for the trials.

4. Finding the Best Frequencies for a Drug

For an accurate pill testing, the FTARS data for a drug has to be analyzed to identify the N frequencies, that can identify the drug best. Finding these fequencies bases on criteria, whose characterice the suitability of a set of N different fequencies.

This characterization will done by a special curve approximation of the response intensities of the different frequencies with respect to the given probes. This curve approximation will be done by minimizing a functional for each frequency tuple.

4.1. Curve Quality.

4.1.1. The General Algorithm. In order to determine the identification ability of a N–tuple of frequencies $\{v_i\}_{i=1}^N \in \mathbb{R}^N$ and test them with m strictly increasing concentrations $\{c_i\}_{i=1}^m \in \mathbb{R}^m$ and get m response vectors $\{f_i\}_{i=1}^m$.

Now a sufficiently smooth parameterized curve $q(\cdot)$, $q : \mathbb{R} \to \mathbb{R}^N$ approximates the data points (respectively the response vectors $\{f_i\}_{i=1}^m$) and projects each f_i, $\forall i = 1, \dots, m$ orthogonally onto $q(\cdot)$ at parameter t_i with the restriction that $t_i < t_{i+1}$ and determines the arc length s_i along the curve from t_1 to t_i afterwards.

The ideal curve $q(\cdot)$ would result in an affine linear relation between arc lengths s_i and concentrations c_i, $\forall i = 1, \dots, m$. The distances of points (s_i, c_i) from the regression line of these points determines the quality of the approximation.

A pseudo algorithm for this quality criterion looks like:

(1) for $i = 1, \dots, m$: project f_i onto $q(t)$, such that for t_i the value $q(t_i)$ is the closest orthogonal projection point of f_i on the curve $q(t)$

(2) for $i = 1, \dots, m$: calculate the arc length s_i along the curve $q(\cdot)$ from t_1 to t_i

(3) determine the parameters a, b of the regression line

$$g(s) = as + b \tag{1}$$

from the data pairs $\{(s_i, c_i)\}_{i=1}^m$

(4) evaluate the quality of the curve $q(\cdot)$ according to the functional

$$F(q, f) := \omega_1 \sum_{i=1}^{m} (g(s_i) - c_i)^2 + \omega_2 \sum_{i=1}^{m} \|q(t_i) - f_i\|^2 + \omega_3 \sum_{i=1}^{m-1} \max(0, t_i - t_{i+1})^2 \tag{2}$$

where ω_1, ω_2 and ω_3 are freely chosen parameters with the restriction that $\omega_3 \gg \omega_1, \omega_2$. The idea of this restriction is that the case $t_i > t_{i+1}$, $i = 1, \dots, m-1$ is to be penalized.

Since F is a convex functional and sufficiently regular, we can find (one of) the best curves for the particular set (v, f, c) in a class by optimizing the curves parameters with respect to F.

The choice of the used curve classes is of big importance: They have to be fast in evaluation and have to allow a good approximation of the response vectors $\{f_i\}_{i=1}^m$.

4.1.2. Projection. Because $q(t) \in C^1(\mathbb{R}^N)$ the tangential vector of $q(t)$ at parameter t^* is $q'(t^*) = \left(q^{(1)\prime}(t^*), \dots, q^{(N)\prime}(t^*)\right)^T$ wherein $q^{(k)\prime}$ denotes the first derivative of $q^{(k)}$ with respect to t. Therefore the projection of the point $\underline{f} \in \mathbb{R}^N$ onto the curve $q(t)$ can be expressed in terms of the inner product $\langle \cdot, \cdot \rangle$ in the Hilbert space $\mathbb{R}^N$

$$\text{Find } t^* \in \mathbb{R} \text{ such that } p(t^*) := \left\langle q'(t^*)\, ,\, \underline{f} - q(t^*) \right\rangle = 0 \ . \tag{3}$$

Providing the first derivative of $p(t)$

$$p'(t) = \left\langle \underline{f} - q(t)\, ,\, q''(t) \right\rangle - \left\langle q'(t)\, ,\, q'(t) \right\rangle \tag{4}$$

allows to solve the non–linear equation $p(t) = 0$ via the Newton iteration

$$t^{(l+1)} := t^{(l)} - \frac{p(t)}{p'(t)} \ . \tag{5}$$

with an appropriate initial guess $t^{(0)}$. We have to be aware that (3) may have non-unique solutions. Therefore this initial guess for the non–linear solution procedure is of great importance.

Solving (3) for all $\underline{f}_i$ will determine the t_i, $i = 1, \ldots m$ from step 1 in the algorithm from §4.1.1.

4.1.3. Arc Length. The formula for the arc length is simply

$$s(t^*) \;:=\; \int_{t_1}^{t^*} \left(\sum_{k=1}^{N} [q^{(k)\prime}(t)]^2 \right)^{0.5} dt \;. \tag{6}$$

A arc–length parameterized curve, i.e., using s instead of t as parameter, is rather hard to achieve and can be done only numerically for the general case. An approach for spline curves is described in [19].

4.1.4. Regression Line. Determining the a, b in the regression line (1) from the given data pairs $\{(s_i, c_i)\}_{i=1}^m \in \mathbb{R}^2$ is equivalent to minimizing the functional

$$\widetilde{F}(a,b) \;:=\; \sum_{i=1}^{m} (g(s_i) - c_i)^2 \overset{(1)}{=} \sum_{i=1}^{m} (a \cdot s_i + b - c_i)^2 \tag{7}$$

with respect to the parameters a, b. Note, that the linear functional $\widetilde{F}$ in (7) is similar (not identical) to the non–linear functional F in (2).

Some simple numerical analysis solves (7) as

$$\begin{pmatrix} a \\ b \end{pmatrix} = \frac{1}{m \sum\limits_{i=1}^{m} s_i^2 - \left(\sum\limits_{i=1}^{m} s_i \right)^2} \begin{pmatrix} m \sum\limits_{i=1}^{m} s_i c_i & - \left(\sum\limits_{i=1}^{m} s_i \right) \left(\sum\limits_{i=1}^{m} c_i \right) \\ \left(\sum\limits_{i=1}^{m} s_i^2 \right) \left(\sum\limits_{i=1}^{m} c_i \right) & - \left(\sum\limits_{i=1}^{m} s_i c_i \right) \left(\sum\limits_{i=1}^{m} s_i \right) \end{pmatrix} \tag{8}$$

4.2. Curve Classes.

4.2.1. Polynomial Curves. Let us assume that the functions $q^{(k)}(t)$ are global polynomial functions of degree p, i.e.

$$q^{(k)}(t) := \sum_{j=0}^{p} a_{kj} \cdot t^j \qquad \forall k = 1, \ldots, N \;. \tag{9}$$

If we use piecewise polynomial function, e.g., cubic splines, then we take into account a similar representation of the curve in which the coefficients a_{kj} will characterize the curve $q(t) \in \mathbb{R}^N$. These coefficients are functions of the given intensities $\underline{f}_i \in \mathbb{R}^N$. The coefficients a_{kj} are unique under the presupposition $q(t = 0) \equiv \underline{f}_1$.

Because neither projection step, nor arc length computation for the algorithm in §4.1.1 can be performed analytically for all polynomials of degree higher than 2, numerical approximation like described in §4.1.2 and §4.1.3 has to be used for projection and arc length computation.

4.2.2. Circular Splines. A circular spline $q\,(t)$ is a parameterized spline curve in $\mathbb{R}^N$, which segments consist of n circular arcs y_i, $i = 1, \ldots, n$ [20].

Each circular arc has a rational Bezier-representation

$$y_i\,(u) = \frac{(1-u)^2 A + 2u\,(1-u)\,\omega B + u^2 C}{(1-u)^2 + 2u\,(1-u)\,\omega + u^2} \qquad u \in [0,1] \;. \tag{10}$$

Hence, $q\left(t\right) = y_k\left(\tilde{u}\right)$ with $k = \lfloor t \rfloor$ and $\tilde{u} = t - k$, where A, B, C and ω describe the arc. All arcs can be joined such that the spline has C^1-regularity.
Projection: $u^* \in [0,1]$ is the parameter of a projection point of $\underline{f} \in \mathbb{R}^N$ on a circular arc y if the relation in (3) is fulfilled for u^*, i.e.,

$$\left\langle y'\left(u^*\right), \underline{f} - \frac{\left(1-u^*\right)^2 A + 2u^*\left(1-u^*\right)\omega B + u^{*2}C}{\left(1-u^*\right)^2 + 2u^*\left(1-u^*\right)\omega + u^{*2}} \right\rangle = 0 \tag{11}$$

Thus, u^* may be computed directly without using iterative methods, since solving (11) for u^* means to find a real zero of a polynomial with degree 4 in the interval $[0,1]$.
Arc Length: Analogously, closed and easily computable formulas for the arc length computation can be derived by taking advantage of the geometrical properties of the spline segments.

4.3. Finding the Best Curve. Formally, determining the best curve is equivalent to minimizing functional F from (2). The functional F is a function of the regression parameters a, b which are functions of all s_i, c_i. The s_i are determined from $q^{(k)}, t_i$ and the latter is a function of curve $q^{(k)}$ and intensities $\underline{f}_i$. The parametrization of $q^{(k)}$ involves finally the curve parameters a_{kj} from §4.2.1, i.e., we consider only the polynomial curves approach in the following. The formulas for circular splines will be similar using the appropriate curve parameters for them.

$$\begin{aligned}
F &:= \sum_{i=1}^{m}(g(s_i) - c_i)^2 \\
& \qquad \text{regression line} \\
&= F(a(\{s_i\}, \{c_i\}),\, b(\{s_i\}, \{c_i\})) \\
& \qquad \text{arc length} \\
&= F(a\left(\left\{s_i\left[\{q^{(k)}\},\, t_i\right]\right\},\, \{c_i\}\right), b(\cdots)) \\
& \qquad \text{projection} \\
&= F(a\left(\left\{s_i\left[\{q^{(k)}\},\, t_i\left(\{q^{(k)}\},\, \underline{f}_i\right)\right]\right\},\, \{c_i\}\right), b(\cdots)) \\
& \qquad \text{parameterization} \\
F(\{a_{kj}\}, \{f_i^{(k)}\}, \{c_i\}) &= F(a\left(\left\{s_i\left[\{a_{kj}\},\, t_i\left(\{a_{kj}\},\, \underline{f}_i\right)\right]\right\},\, \{c_i\}\right), b(\cdots))
\end{aligned}$$

Determining the best curve for an N-tuple of frequencies means solving the optimization problem

$$\min_{a_{kj}} F(\{a_{kj}\}, \{\underline{f}_i\}, \{c_i\}) \qquad \forall i = 1, \ldots, m \tag{12}$$

which can be expressed as non–linear equation

$$\nabla_{a_{kj}} F(\{a_{kj}\}, \{\underline{f}_i\}, \{c_i\}) = 0 \tag{13}$$

with respect to the design variables a_{kj}, $k = 1, \ldots, N$, $j = 0, \ldots, p$.

Solving (13) directly cannot be done analytically because of the non–linear equation (3) the gradient cannot be provided analytically. Instead, we tackle the optimization problem (12) with standard SQP methods [21, 22]. The optimizer is based on a Quasi–Newton approximation of the Hessian using a modified BFGS update formula following [23] in order to avoid the need for Hessian information of the objective.

Algorithm 1 Quasi–Newton optimization with BFGS update

```
Set i=0
Make initial guess for the design parameters $x_i$.
Set the approximation of the hessian $H_i = I$.
while convergence criterion is not fulfilled do
  Calculate descent direction $d_i$ from $H_i d_i = -\nabla_{x_i} F(x_i)$
  Perform line search to get a step size $\alpha_i$ with sufficient descent
  Set $x_{i+1} = x_i + \alpha_i d_i$
  Set $H_{i+1}$ according to the BFGS formula
  i=i+1
end while
```

4.4. Implementational Aspects.

Parallelization. For k possible frequencies, the number of the N–tuples that have to be tested and hence is the number of curve approximations that have to be performed is $\binom{k}{N}$. This means finding the best frequencies is a very computational task. Therefore parallelization of the task is required. Basically, finding the best curve for a frequency tuple (and hence evaluate the quality of this tuple) is a sequential problem for one CPU, respectively for one CPU core. Therefore the parallelization approach is to have a distinguished master process and many slave processes.

The master process takes care of the data handling (supplies all other processes with data sets) and also keeps track of the quality of each data set, while the slave processes receive one or multiple data set from the master, find the best curve for the data sets (by performing the algorithm above), send the result back to the master process and evaluate the new data sets received from the master.

Optimization. While the optimization algorithm 1 with propper line search can guarantee convergence rates, it prove to be unsuitable for the structure of the functional F. The reason for this is the steepness of F at certain points, that leads to problems with common line search strategies, especially when single–precision data types are used for computation.

Thus, a very naive approach appears to be useful: For line search we use backtracking until any descent is achieved. In the step size control any step length is valid, as long as there is any descent with respect to a step length of 0. This approach does not guarantee any convergence rates, but prove to be suitable for F.

5. Identification and Computational Results

We identified several toll-manufactured drugs (aspirin, acetaminophen, D-tagatose, ibuprofen, vitamin C, and vitamin B) in a controlled lab environment using ISP acoustic waveforms composed of 10, 100, and 1000 frequencies. We found that just the top 10 frequencies properly classified each pill. Intra-cluster distances were calculated to be less than 3 multidimensional standard deviations (MSD) for each pill type. The average accuracy of prediction was 98.47, 97.45 and 95.41 percent for the 10, 100 and 1000 frequency component acoustic waveforms respectively [18].

Computing the libraries for the ISP-ARS devices requires a large number of independent calculations. We used a cluster made up of Intel Core i7-920 running at 2.67GHz and 6GB DDR3 RAM. Each potential frequency was computed using 8 cores. We used 10,000 point datasets and 9-way parallelism (the ninth is the controlling process and uses minimal time).

Using the g++ compiler, run times per frequency calculation were approximately 160, 260, and 321 seconds for third through fifth degree arc lengths used to find the best frequencies.

We can deliver an infinite number of acoustic spectra, but that defeats the creation of a small, embedded DDDAS device that is useful in itself. Instead we choose a small number of spectra, which changes slightly over time based on environmental and personnel factors. However, we compute on as many processors as we can for a set period of wall clock time. For real production of pills, we need to recalibrate the libraries once per work shift. We can predict how long we need to compute based on timings like above and the number of processors we can access on a supercoputer. Alternately, we can predict how many processors we need based on how much time we are allowed.

6. Conclusions

We have described a multiscale prototype DDDAS to identify defective, inadequately low quality, or mislabeled pills. ISP-ARS can differentiate between different types of pills. The results are preliminary and much more research and development is necessary to produce systems that can be deployed on pharmaceutical manufacturing lines with government approvals.

Acknowledgements

This research was supported in part by NSF grants CNS-1018072 and CNS-101807 and Award No. KUS-C1-016-04, made by King Abdullah University of Science and Technology (KAUST) and the Austrian Science Fund FWF under grant SFB F032.

References

[1] http://www.fda.gov/diabetes/qna.htm

[2] http://www.nlm.nih.gov/medlineplus/diabetes.html

[3] http://www.diabetes.org/diabetes-statistics/cost-of-diabetes-in-us.jsp

[4] http://www.fda.gov/ohrms/dockets/ac/02/briefing/3869B1_08_woodcock/sld001.htm

[5] http://www.dddas.org

[6] Baldridge, K., Biros, G., Chaturvedi, A., Douglas, C.C., Parashar, M., How, J., Saltz, J., Seidel, E., Sussman, A.: January 2006 DDDAS Workshop Report, National Science Foundation, (2006)

[7] Douglas, C.C., Deshmukh, A., Ball, M., Ewing, R.E., Johnson, C.R., Kesselman, C., Lee, C., Powell, W., Sharpley, R.: Dynamical data driven application systems: Creating a dynamic and symbiotic coupling of application/simulations with measurements/ experiments, National Science Foundation, Arlington, VA, (2000)

[8] 2007 Dynamic Data-Driven Application Workshop, F. Darema, ed., in Computational Science ICCS 2007: 7th International Conference, Beijing, P.R. China, May 28-31, 2007, Proccedings, Part I, Y. Shi, G. D. van Albada, P. M. A. Sloot, and J. J. Dongarra (eds.), Springer-Verlag, Heidelberg, Lecture Notes in Computer Science series, vol. 4487, pp. 955-1245, (2007)

[9] 2009 Dynamic data-driven application systems - DDDAS 2008, C.C. Douglas, ed., in Computational Science ICCS 2008: 8th International Conference, May 25-27, 2009, Proceedings, Part II, Gabrielle Allen, Jaroslaw Nabrzyski, Edward Seidel, Geert Dick van Albada, Jack J. Dongarra, and Peter M. A. Sloot (eds.), Lecture Notes in Computer Science series, vol. 5545, Springer-Verlag, Heidelberg, pp. 445-509, (2009)

[10] Dai, B., Urbas, A., Douglas, C.C., Lodder, R.A.: Molecular factor computing for predictive spectroscopy. Pharm Res. 8, pp.1441-144, (2007)

[11] Beadle, J.R., Saunders, J.P., Wajda, Jr, T.J.: Process for Manufacturing Tagatose, Biospherics Incorporated, (US patent 5,002,612). Available from URL: http://www.uspto.gov/patft/index.html, (1991)

[12] Beadle, J.R., Saunders, J.P., Wajda, Jr, T.J.: Process for Manufacturing Tagatose, Biospherics Incorporated, (US patent 5,078,796). Available from URL: http://www.uspto.gov/patft/index.html, (1992)

[13] Lu, Y., Levin, G.V., Donner, T.W.: Tagatose, a new antidiabetic and obesity control drug, Diabetes Obes Metab. 10, pp. 109-134, (2008)

[14] The MathWorks, Simulink and the Communications Toolbox, http://www.mathworks.com

[15] Woodcock, J., US Food and Drug Administration. Available: http://www.fda.gov/ohrms/dockets/ac/02/briefing/3869B1_08_woodcock/sld001.htm

[16] Parashar, M., Matossian, V., Klie, H., Thomas, S.G., Wheeler, M., Kurc, T., Saltz, J., Versteeg, R.: Towards Dynamic Data-Driven Management of the Ruby Gulch Waste Repository, in Computational Science - ICCS 2006: 6th International Conference, Reading, UK, May 28-31, 2006, Proceedings, Part III, Vassil N. Alexandrov, Geert Dick van Albada, Peter M.A. Sloot, Jack J. Dongarra (eds.), Lecture Notes in Computer Science series, vol. 3993, Springer-Verlag Heidelberg, pp. 384-392, (2006)

[17] Medendorp, J.P., Fackler, J.A., Douglas, C.C., Lodder, R.A.: Integrated sensing and processing acoustic resonance spectrometry (ISP-ARS) for sample Classification, J. Pharm. Innov, 2, pp. 125-134, (2007)

[18] Hannel, T., Link, D., Lodder, R.A.: ISP-ARS in discriminating Tagatose and other toll manufactured drugs, ISPE Journal of Pharmaceutical Innovation, 3, pp. 152-160, (2008)

[19] Wang, H., Kearney, J., Atkinson, K.: Arc-length parameterized spline curves for real-time simulation. In: T. Lyche, M. Mazure, L. Schumaker (eds.) Proceedings of the 5-th International Conference on Curves and Surfaces, San Malo 2002, pp. 387396. Nashboro Press, Brentwood, TN (2003)

[20] Song, X., Aigner, M., Chen, F., Jüttler, B.: Circular spline tting using an evolution process. Journal of Computational and Applied Mathematics 231, 423433 (2009)

[21] Gill, P.E., Murray, W., Wright, M.H.: Practical Optimization. Academic Press, London-San Diego-New York (1981)

[22] Nocedal, J., Wright, S.: Numerical Optimization. Springer Series in Operations Reasearch. Springer, New York (1999)

[23] Powell, M.J.D.: A fast algorithm for nonlinear constrained optimization calculations. In: G.A. Watson (ed.) Numerical Analysis, no. 630 in Lecture Notes in Mathematics. Springer, Berlin (1978)

School of Energy Resources and Department of Mathematics, University of Wyoming, Laramie, WY 82071-3036 USA
E-mail: craig.c.douglas@gmail.com and ldeng@uwyo.edu

Department of Mathematics, Texas A&M University, College Station, TX USA
E-mail: efendiev@math.tamu.edu

Institute for Mathematics and Scientific Computing, University of Graz, A-8010 Graz, Austria
E-mail: gundolf.haase@uni-graz.at and andreas.kucher@gmail.com

Department of Chemistry, University of Kentucky, Lexington, KY 40506 USA
E-mail: lodder@uky.edu