

## Global Journal of Advanced Engineering Technologies and Sciences

### DERIVATION OF THE FOURIER TRANSFORM REPRESENTATION OF A NONPERIODIC SIGNAL

Mohsen Hamidi<sup>1</sup>

<sup>1</sup> Department of Electrical Engineering, Mahabad Branch, Islamic Azad University, Mahabad, Iran

#### Abstract

It is a difficult task to balance societal need for development and the need to sustain the environment as a life supporting system of the planet. Engineers and scientists have accept the current and future challenges in order to ensure the sustainability of both the environment and the society. Two major changes are needed in the training and practices of scientists and engineers. First, the need to adopt advanced technologies as the basic knowledge platform in solving environmental related problems. Second, the need to understand and embrace multi-, inter- and trans-disciplinary approaches in solving current and future problems. Most important of all, the solutions provided must not only be feasible from the technological and economics point of view but it must also address the psychological and emotional well-being of the society.

**Keyword:** Modulation, bandwidth, coding, CDM, Synchronization

#### I Introduction and Related Works

Among them, the wearable sensors enable to capture the high velocities, which occur during the critical phase, and the main objective of a communication system is the reliable transfer of information over a channel. Typically the information is represented by audio or video signals, though one may easily postulate other signals, e.g., chemical, temperature, and of course text, i.e., the written word which you are now reading. Regardless of how the message signals originate they are, by their nature, best modelled as a random signal (or process). This is due to the fact that any signal that conveys information must have some uncertainty in it. Otherwise its transmission would be of no interest to the receiver, indeed the message would be quite boring (known known so to speak). Further, when a message signal is transmitted through a channel it is inevitably distorted or corrupted due to channel imperfections. Again the corrupting influences such as the addition of the ever present thermal noise in electronic components, the multipath fading experienced in wireless communications are unpredictable in nature and again best modelled as nondeterministic signals or random processes. However, in communication systems one also utilizes signals that are deterministic, i.e., completely determined and therefore predictable or non-random. The simplest example is perhaps the

carrier used by AM or FM analog modulation. Another common example is the use of test signals to probe a channel's characteristics. Channel imperfections can also be modeled as deterministic phenomena: these include linear and nonlinear distortion, intersymbol interference in bandlimited channels, etc. This chapter looks at the characterization and analysis of deterministic signals. Random signals are treated in the next chapter. It should be mentioned at the outset that in certain situations whether a signal is considered deterministic or random is in the eyes of the viewer. Only experience and application can answer this question. The chapter reviews deterministic signal concepts such as Fourier series, Fourier transform, energy/power spectra, auto and cross correlation operations. Most of the concepts discussed carry over to random signals. Though intended to be self-contained it is expected that the reader has been exposed to these concepts in other undergraduate courses, hence the treatment is presented succinctly A deterministic signals is a defined, or completely specified, function of an independent variable. The independent variable is invariably taken to be time (seconds) but it may be some other independent variable, e.g., position, heat input (joules), etc. Typically the signal is represented by a graph or analytically by an equation; but other possible representations are a table of values, a

statement in English, an algebraic or differential equation, or some combination of the above. The most common representation is that of an equation, several

Examples of which are given below along with their graphical representations.

(a)  $s(t) =$

$1, t \geq 0$

$0, t < 0$

. Known as the unit step function and often denoted by  $u(t)$ .

(b)  $s(t) = e^{-at}u(t)$ ,  $a > 0$ . A decaying exponential function

(c)  $s(t) = e^{-a|t|}$ ,  $a > 0$ . Also a decaying exponential. It is an even function (also known as a tent function).

(d)  $s(t) = V \cos(2\pi fct)$ . A periodic sinusoid, whose fundamental period is  $1/fc$  (seconds).

(e)  $s(t) = V \cos(2\pi fct)$

$u(t + T/2) - u(t - T/2)$

. A tone burst of duration  $T$  (seconds).

(f)  $s(t) = \sum_{k=-\infty}^{\infty} p(t - kT)$ , where  $p(t)$  is an arbitrary signal (pulse) for  $0 < t \leq T$  and equal to zero elsewhere. This is a train of delayed pulses.

(g)  $s(t) = m(t) \cos(2\pi fct + dm(t)/dt)$ , where  $m(t) = A \sin(2\pi f_m t)$ ,  $f_m \ll f_c$ . An amplitude/phase modulated signal.

(h)  $s(t) = g(t - t_0)$ , where  $g(t)$  is any defined signal, perhaps one of the above. This is simply a time shift of  $g(t)$  by  $t_0$  seconds – to the right if  $t_0$  is positive and to the left if negative.

Signals (a)–(f) are represented graphically in Figure 2.1. Though signals can be represented graphically or analytically in the time domain, it is common to seek other representations which allow one to gain insight into the signal characteristics and/or to deal with them more successfully. The most ubiquitous representation is a frequency domain one via either the Fourier series (for periodic signals) or the Fourier transform (for aperiodic signals). Periodic signals have more structure than aperiodic signals and since their representation provides the stepping stone to aperiodic signal representation, they are discussed next. To

proceed it seems advisable to establish a procedure to determine the coefficients of the series expansion. If the coefficients  $A_k, B_k$  can be found then  $C_k, \theta_k$ , and  $D_k$  are known. Similarly, knowledge of  $D_k$  allows one to determine  $A_k, B_k$  or  $C_k, \theta_k$ . The coefficients  $A_k,$

$B_k, D_k$  are readily determined by using the orthogonality property of harmonic sinusoids; indeed that is why the frequencies of the sinusoids were chosen as they were. Orthogonality is an important enough property to warrant a more general definition and discussion. Note that  $D_m$  needs to be calculated only for  $m = 0, 1, 2, \dots$ . Then the fact that  $D_{-m} = D_m^*$  is used for negative  $m$ . Though simple enough in principle the integrations of (2.10), (2.11) and the resultant algebra can be quite tedious in practice. Some of this tedium can be alleviated, but alas never completely eliminated, by using the six properties that are presented and discussed next. Judicious application of these properties along with exploitation of signal symmetries can in some instances significantly reduce the effort. Even if the straightforward approach is taken to determine the coefficients, the properties may be used to provide a check on one's work.

An interesting object (i.e. the foreground) will be extracted from the background of each video frame in a video clip. This detection technique is called background subtraction. This approach detects the foreground by thresholding the difference between the current frame and the modelled background in pixel by pixel process. After blobbing and smoothing the object in Filter module, this result will be tracked by 2D modeling such as point tracking, kernel tracking (rectangle, ellipse, skeleton...), or silhouette tracking. Then, the feature extractions to understand object gestures are calculated based on one of these modeling. The problems are that these features must encapsulate unique characteristics for the same activity but acting by different people. In order to avoid misdetection and false alarms for this system, it not only depends on the techniques but also confronts some challenges such as

- Dynamic background: in some cases there are many moving objects in an indicated scene and how to extract the exact object.
- Brightness: The changes of light intensity at different times of the day or the suddenly turning of lights also affect the background processing.
- Occlusions: are often considered as artifacts, undesirable in many image motion computations. We

have argued that occlusions show the related important information about the camera motion (position and angle of camera) and scene structure (static or dynamic environment; brightness, colour costume of objects).

- Static object: when the object is sleeping, sitting or walking calmly, it is easy to mistake the extraction of object.

After the object features are tracked and extracted, the problem of the system has to understand the meaning of the object actions through their features in the recognition block.

## II. Implementation of Fall Detection Application

Recently, a video processing framework called OpenCV is regularly developed on an open source library for computer vision of Intel. However, how to implement video applications with OpenCV on a hardware platform are really big challenges. In the work of Floris Driessen [8], he proposed the combination of embedded processors and customized accelerators on heterogeneous computation platform - the Zynq-7000 all programmable system-on-chip -, which offers a high-end embedded processor combined with field programmable gate array (FPGA) based reconfigurable logic. Moreover, another fall detection system which was implemented on Terasic's DE2-115 development board including Altera Cyclone IV (EP4CE115) FPGA device, a 5 megapixels CMOS camera sensor and a LCD touch panel. This system was also designed with highly exploitation of the parallel and pipeline architecture of the FPGA [9].

Shimmer technology and applied the orthogonal matching pursuit (OMP) algorithm for advanced data compression was shown in [10]. This system has been simulated and implemented on the Virtex-5 and Zynq 7 (FPGA) using Vivado High Level Synthesis tool. It is used to estimate the area, power and computation time for the detection of falls with different scenarios. In the other side, the combination of Kinect and Wireless Accelerometer is used to extract object of images in a dark room [11]. This system was implemented on the Panda Board ES with real-time indication.

In the next section, the research objective is mentioned. Fall Detection application will be described in Section III with four steps: object segmentation, filter, feature extractions and recognition. In Section IV an insightful experiment

of implementation and evaluation is described. Finally, Section V contains the conclusions of this paper.

## III. Research Objective

To deploy applications of embedded systems on heterogeneous platforms which contain various computation units such as Field Programmable Gate Array (FPGA), microprocessor and graphics processing unit GPU, we need to determine a specific activity of design phase. Then, the designers are going to profile the partition for software (SW) or hardware (HW) implementation.

In common practice, deployment decisions are taken at an early stage of the design phase, and it branches into two separated flows: HW and SW design flows. Then, during the implementation phase they evolve separately until the final integration. In this scenario, the design phase is affected by issues such as HW or SW flow interruptions, redesigns and unplanned iterations which negatively impact the overall development process in terms of efficiency, quality and costs, and the system lifecycle. Especially, this method is time-consuming to separate implementation on HW and SW for whole application.

In our study, we extract the execution time, power consumption of whole fall detection application which is deployed on ARM processor of Zynq-7000 AP SoC platforms. After that, the modules in Fall Detection which take the most execution time will be implemented on HW (with FPGAs).

## IV. Fall Detection Application

### 1. OBJECT SEGMENTATION

Object segmentation block is responsible for detecting and distinguishing between moving objects and the rest of the frame which is called background. Background subtraction method is applied in this study. A pixel is marked as Foreground If  $|I_i - B_i| > \tau$ , (1)

Where  $I_i$  is current video frame;  $B_i$  is modelled background frame and  $\tau$  is a "predefined" threshold. The updated background is estimated as:  $B_{i+1} = \alpha I_i + (1 - \alpha) B_i$  (2) Where  $\alpha$  is kept small to prevent artificial "tails" forming.

### 2. FILTER

There are some supporting methods to improve the quality of image from the object binary image such as Morphology Mathematic (MM), Edge Detection Filter

(Sobel, Canny, Prewit Filter). In this module the object will be smoothed, blobbed and the noise will be removed.

Morphology Mathematic (MM) is a technique for the analysis and processing of geometrical structures. It is also used in morphological image processing by transforming images according to the characterizations of object. Some of MMs are dilation, erosion, opening, closing or the combination of these.

Sobel edge detection algorithm is the most commonly used techniques in image processing for edge detection [12]. In this paper two types of Sobel operators, horizontal, vertical, were used. The operator calculates the gradient of the image intensity at each point, giving the direction of the largest possible increase from light to dark and the rate of change in that direction. The Sobel kernels are given by

Here the kernel Gx is sensitive to changes in the x direction, i.e. edges that run vertically, or have a vertical component. Similarly, the kernel Gy is sensitive to changes in y direction, i.e. edges that run horizontally, or have a horizontal component. The two gradients computed at each pixel (Gx and Gy) by convolving with above two kernels can be regarded as the x and y components of gradient vector. This vector is oriented along the direction of change, normal to the direction in which the edge runs. Gradient magnitude and direction are given by:

$$G = \sqrt{G_x^2 + G_y^2}$$

An approximate magnitude is computed using:

$$|G| = |G_x| + |G_y| \quad (5)$$

The angle of orientation of the edge (relative to the pixel grid) giving rise to the spatial gradient is given by:

$$\theta = \arctan \frac{G_y}{G_x}$$

### 3. FEATURE EXTRACTION

ELLIPSE MODEL is usually used for tracking object because it is easy to fit an ellipse around the object. In this study, there are three following parameters needed to set an ellipse: Centroid of ellipse, Vertical angle of the object, Major and Minor Axis of the Object

Centroid of ellipse: for each binary frame, centroid coordinate of an ellipse O (O<sub>x</sub>, O<sub>y</sub>) is determined: Abscissa (O<sub>x</sub>) and ordinate (O<sub>y</sub>) are average of the entire x coordinates and the entire y coordinates of the white pixels.

Vertical angle of the object: after determining the centroid coordinate, the system calculates the angle between major axis ellipse and horizontal line (θ or current angle is calculated by Equation 7).

$$\theta = \frac{1}{2} \cdot \arctan \frac{\sum \sum x \cdot y \cdot P(i,j)}{\sum \sum x^2 \cdot P(i,j)} \quad (7)$$

Where i, j is position of pixel (i=1...width of frame, j=1...height of frame)  
 x = i - O<sub>x</sub> and y = j - O<sub>y</sub> (O<sub>x</sub>, O<sub>y</sub>: position of centroid), and P (i, j): value of pixel (i, j).

### Major and Minor Axis of the Object

a and b are respectively semi-major axis and semi-minor axis of the ellipse. d1 and d2 are distances from O to O1(x1, y1) and O2(x2, y2), respectively. O1 and O2 are calculated: (x1, y1) and (x2, y2) coordinates are the

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

average of the entire x coordinate and y coordinate of the white pixels W (W<sub>x</sub>, W<sub>y</sub>) satisfying the 2 following conditions:

y-ordinate of these white pixels (W<sub>y</sub>) are smaller than the y-coordinate of centroid (O<sub>y</sub>).

These W are in the limited angle so that  $|\theta - \theta_0| \leq 60^\circ$  for O1 and  $|\theta - \theta_0| \leq 60^\circ$  for O2. Finally, major and minor axes are calculated: a = 2d1, b = 2d2.

### FEATURE EXTRACTION

Current angle: cf. Equation 7. Coefficient of motion (Cmotion) At the same position versus time gray pixels are white pixels of the previous frame. Brightness of the gray pixel increases versus the time frame by frame containing white pixel at the same position. The gray frame uses to determine the object's motion rate also known as Motion History Imaging (MHI). The motion coefficient of object Cmotion is determined by:

$$C_{motion} = \frac{\text{Gray pixel}}{\text{Gray pixel} + \text{White pixel}}$$

Where "White pixel" is the number of white pixels; "Gray pixel" is the number of gray pixels. Cmotion's value must be in range of [0, 1]. Deviation of the angle (Ctheta) Ctheta is standard deviation of 15 angles θ from 15 successive frames. Ctheta is usually higher when a fall is occurring. Eccentricity at current frame is computed by:

$$e = \frac{c}{a} \quad (9)$$

Where e: eccentricity; a, b: semi-major and semi-minor axis of the ellipse; e is smaller when direct fall happens.

Deviation of the centroid (Ccentroid) is defined as standard deviation of 15 ordinates from 15 successive frames. Ccentroid rapidly decreases when the fall occurs.

#### 4. RECOGNITION BASED ON TEMPLATE MATCHING

Based on the direction of falls and the type of falls, four models have been built to detect falling accidents. The first model is face fall, in this case Cmotion, Ctheta, Ccentroid is high but Theta is low. The second model is cross fall, with high Cmotion and Ctheta, whilst Theta, Ccentroid and Eccentricity have medium values. In the next model, the victim fall in the perpendicular direction to the camera, consequently Theta is almost constant, Cmotion is in average, the Eccentricity is low while Ccentroid is quite high. The other cases are included in the last model. The features are combined with each other

depending on the fall models, and the thresholds are selected from the survey of training videos.

#### V. Implementation & Evaluation

We design this system in High Level Languages specified in C/C++ integrated OpenCV and cross-compiled along with libraries which implement the communication Application Programming Interfaces (APIs) and runtime layer using gcc/g++ tool chains. These tool chains generate an .elf file which is downloaded to the processor ARM Cortex A9 on

**Train set:** Clear data consists of videos which have stable background. These videos are captured in a small room under good brightness condition. The object is not obscured by furniture in the room. Train set contains 21 videos of fall and 26 videos of daily activities.

**Test set:** Contents and activities in the video clips for testing are basically performed similar to the ones for training, just a small difference of environment condition. In each clip, there is only one object with stable background and include 21 fall videos and the

Zynq platform supported by SDK tools. Moreover, this system is emerged the power consumption; estimated the execution time of each frame from input data on this processor. The frequency of this core is set up at 666 MHz.

The execution time is extracted from available time.h library. The measurements of power and thermal are taken by the Fusion Digital Power Designer GUI. The TI USB Adapter is connected to controller Power Management Bus (PMBus) on Zynq platform and to PC for displaying measurement results, as shown in Fig. 3.

In this study, the input video is recorded by the Camera Web Cam -Philips SPC 900NC PC that is mounted on the wall at the distance of 3m from the floor. The data are captured with both resolution 320x240 pixels and 680x360 pixels.

DATABASE: the fall direction is subdivided into three basic directions

*Direct fall:* object falls face to the camera.

*Cross fall:* occurs when the object falls cross the camera.

*Side fall:* the object perpendicularly falls to both sides of the camera.

In terms of non-fall videos, usual activities which can be misrecognized with fall action such as lying, sitting, creeping, bending are also classified into three directions above. In this study, we create two dataset (as shown in Table 1):

Table 1 Classification of videos

|          | Video            | Database |      |     |     |
|----------|------------------|----------|------|-----|-----|
|          |                  | Train    | Test | Sum | Sum |
| Fall     | Side-Fall (F1)   | 7        | 7    | 14  | 42  |
|          | Direct-Fall (F2) | 8        | 6    | 14  |     |
|          | Cross-Fall(F3)   | 6        | 8    | 14  |     |
| Non Fall | Bending(N1)      | 6        | 8    | 14  | 59  |
|          | Lying (N2)       | 5        | 8    | 13  |     |
|          | Creeping (N3)    | 9        | 8    | 17  |     |
|          | Sitting(N4)      | 6        | 9    | 15  |     |
| Sum      |                  | 47       | 54   | 101 | 101 |

#### CLASSIFYING EVALUATION:

ROC (Receiver Operating Characteristics) is one of the methods to evaluate the efficient and accuracy of a system by calculating the Precision (PR), Recall (RC) and Accuracy (Acc), see in the Equation 6.

$$PR = \frac{TP}{TP + FP} ; RC = \frac{TP}{TP + FN} \quad Acc = \frac{TP + TN}{TP + TN + FP + FN} \quad (6)$$



here TP, TN, FN, and FP are defined as follows:

True positives (TP): amount of fall actions which are correctly classified as fall.

False positives (FP): amount of non-fall actions which are wrongly considered to be fall.

False negatives (FN): amount of fall actions which are wrongly rejected and classified as non-fall actions.

True negative (TN): amount of non-fall actions which are correctly classified as non-fall.

**CONFUSION MATRIX**

A confusion matrix gives information about actual and predicted classifications done by a classification system. Performance of such systems

From the confusion matrix, the Recall, Precision and Accuracy are calculated and shown in Fig. 4. The result of clear data in Train set is higher than Test set in all Recall, Precision, and Accuracy. The reason is that Template Matching uses “hard threshold” and the combination of features is quite simple to detect a fall event. Four models of the fall are not enough to describe all falls may occur.

**2. THE MEASUREMENT**

As shown in the Fig 5, the mean of total execution time of Fall Detection application is approximately 0.107s/frame. The Frame Filter module based on Morphology Filter takes around 2/3 times of total execution time. The similar observation has been obtained when using higher resolution input video of 680x360 pixels. In which the execution time is 0.221ms/frame for Frame Filter and 0.3s/frame for total execution time.

In this case, the measured power consumption of whole Fall Detection application is closed to 0.403W. Therefore, the energy per frame is multiplied by the power consumption (P) and total execution time (T) as following:  $E = P * T = 0.403 * 0.107 = 0.043$  (J/frame) As the result of this experiment, the frame rate of this system is calculated by:  $\text{Frame rate} = 1 / 0.107 = 9.3$  fps

It is found that the over all of this system does not keep on operation at 24 frames per second. Thus, this parameter will have an effect on the recognition ability of this system. That is also big challenge in

is commonly evaluated using the data in the matrix. The following Table 2 shows the confusion matrix for two classes which are classified FALL or NON FALL for both dataset of Train and Test which are implemented on ARM Cortex A9 of Zynq-7000 AP SoC platform.

Table 2 Confusion matrix

|          | Video    | Train    |          | Test |          |   |
|----------|----------|----------|----------|------|----------|---|
|          |          | Fall     | Non Fall | Fall | Non.Fall |   |
|          |          | Database | Fall     | F1   | 7        | 0 |
| F2       | 7        |          |          | 1    | 5        | 1 |
| F3       | 5        |          |          | 1    | 5        | 3 |
| Non Fall | Non Fall | N1       | 1        | 5    | 1        | 7 |
|          |          | N2       | 1        | 4    | 2        | 6 |
|          |          | N3       | 2        | 7    | 1        | 7 |
|          |          | N4       | 1        | 5    | 2        | 7 |

video design to get the reasonable precision, accuracy, performance when we change the offline procession by online one.

**3. HW IMPLEMENTATION**

The Zynq@-7000 family is based on the Xilinx All Programmable SoC architecture. These products integrate a featurerich dual-core ARM® Cortex™-A9 based processing system (PS) and 28 nm Xilinx programmable logic (PL) in a single device. The ARM Cortex-A9 CPUs are the heart of the PS and also include on-chip memory, external memory interfaces, and a rich set of peripheral connectivity interfaces [13]. As a result, the Zynq-7000 AP SoCs are able to serve a wide range of applications including video application.

As discuss in the section II and from the observed results in Fig. 5, the Frame Filter is chosen for implementation on HW. In order to reduce the execution time, the Sobel Filter is replaced by the Morphology Filter. In Sobel Filter, two types of Sobel operators, horizontal and vertical, were used. The operator calculated the gradient of the image intensity at each point, giving the direction of the largest possible increase from light to dark and the rate of changein that direction [14]. By this way, the power and execution time of this module are estimated by using the Vivado\_HLS tool. From the power consumption and execution time shown in Table 3, the energy consumption per frame is calculated as

$$(*) \quad E = P \cdot T = 1.65 \cdot 10^{-3} \text{ (J/frame)}$$

$$(**) \quad E = P \cdot T = 1.158 \cdot 10^{-3} \text{ (J/frame)}$$

With the same function for Frame Filter, the Sobel Filter implemented on HW spends forty times less energy comparison with Morphology Filter implemented on ARM processor.

After that the previous experiment is performed to select which modules or algorithms are candidates for HW or SW implementations. Therefore, the HW/SW architecture is proposed for this Fall Detection application, in which Frame Filter based on Sobel Filter is selected to obtain the better performance and low power consumption.

**Table 3 Power and execution time of frame filter on HW**

| Frame Filter | Power Consumption (mW) | Execution Time (ms) |
|--------------|------------------------|---------------------|
| 680x360 (*)  | 198                    | 8.34                |
| 320x240(**)  | 193                    | 6.15                |

## V. Conclusion

When one models a physical signal, one looks for a model that both captures the essence of the signal and also simplifies subsequent analysis. One consequence is that discontinuities are introduced. Perhaps the square pulse is the simplest example of this. Differentiation of discontinuities leads to the concept of the impulse or delta function. The impulse function, though strictly speaking a mathematical fiction, is of considerable importance in signal and system analysis. Before presenting the examples, the impulse function is discussed.

## References

- I. [1] O. Petit and C. Baron, "Integrated water resources management: From general principles to its implementation by the state. The case of Burkina Faso," *In Natural Resources Forum*, vol. 33, pp. 49–59, 2009.
- II. [2] S. Pollard and D. Du Toit, "Integrated water resource management in complex systems: How the catchment management strategies seek to achieve sustainability and equity in water resources in South Africa," *Water SA*, vol. 34, pp. 671-679, 2008.
- III. [3] S. R. Lim, S. Suh, J. H. Kim, and H. S. Park, "Urban water infrastructure optimization to reduce environmental impacts and costs," *Journal of Environmental Management*, vol. 91, pp. 630-637, 2010.
- IV. [4] S. Lanini, N. Courtois, F. Giraud, V. Petit, and J. Rinaudo, "Socio-hydrosystem modelling for integrated water-resources management-the Hérault catchment case study, Southern France," *Environmental Modelling & Software*, vol. 19, pp. 1011-1019, 2004.
- V. [5] Y. A. Mylopoulos, A. K. Mentis, and I. Theodossiou, "Modelling residential water demand using household data: A cubic approach," *Water International*, vol. 29, pp. 105-113, 2004.
- VI. [6] M. Yurdusev and P. O'connell, "Environmentally-sensitive water resources planning," *Water Resources Management*, vol. 19, pp. 375-397, 2005.
- VII. [7] L. J. Pearson, A. Coggan, W. Proctor, and T. F. Smith, "A sustainable decision support framework for urban water management," *Water Resources Management*, vol. 24, pp. 363-376, 2010.