Data Storage Using Mathematical Model

Binoy Kurikaparambil Revi

Independent Researcher
binoyrevi@live.com

Abstract:
Data is precious and critical in today’s world of IOT and Big data. When it comes to data storage it has become even more expensive with the new technologies. Data Storage using Mathematical Models is a scientific technique that can help a fairly predictable system to store the data in a very compact form. This technique can also combine with the traditional data storage mechanism to provide a Hybrid Data Storage using Mathematical Model. The technique is very useful in storing data such as data for IOT and big data applications where the data can be considered as a continuous time model or discrete time model or even discretized versions of continuous time model.

A Typical Use Case

Let’s consider a very common use case that is generally used to run analytical, monitoring and control functions. Consider a system that uses data from so many nearby and remote locations to perform monitor and control functions. In this scenario, data from sensors and measurement devices are transmitted to a data hub or computer application is heavily dependent on the time the data is captured. Assume that the data is stored in the disk and used by a software application to monitor certain parameters and issue control signals to the control unit.

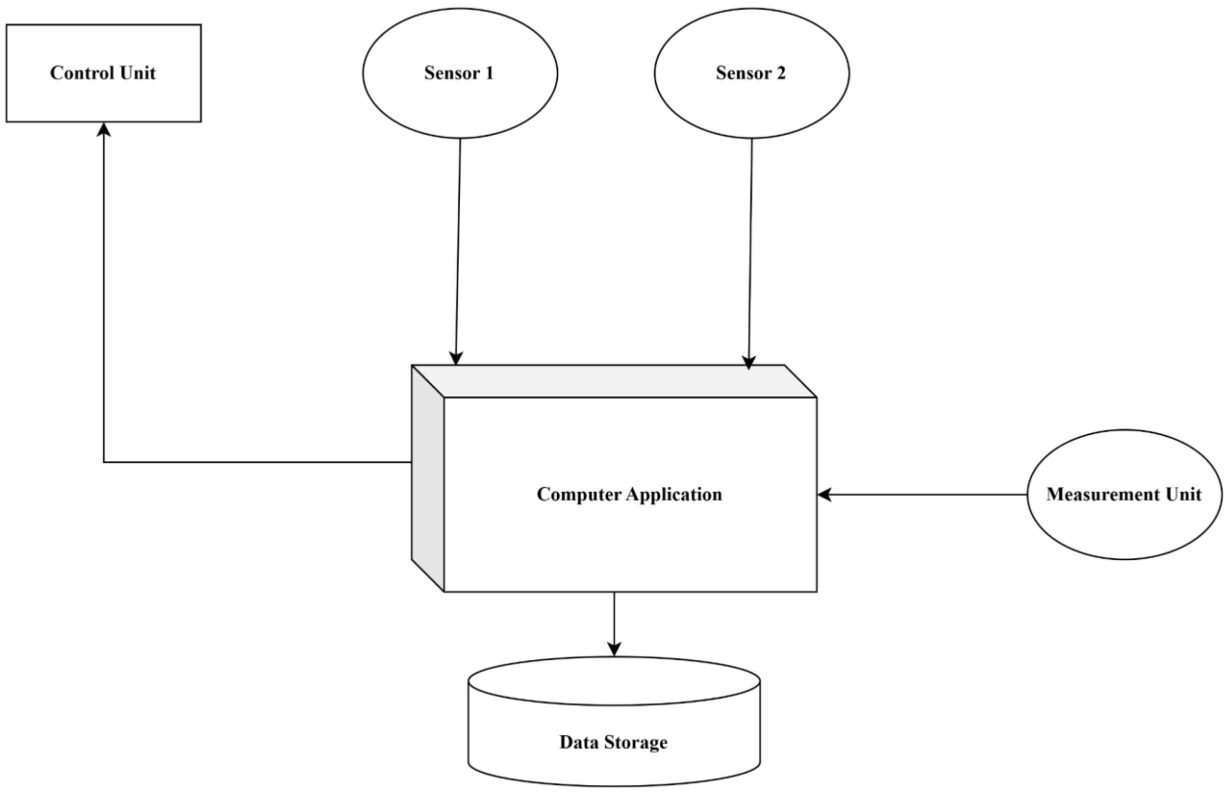

Figure 1: A Simple Measurement and Control System

Data Analysis

Let's analyze data that is received by the system and stored into this digital storage. I created a data chart for this analysis as below.

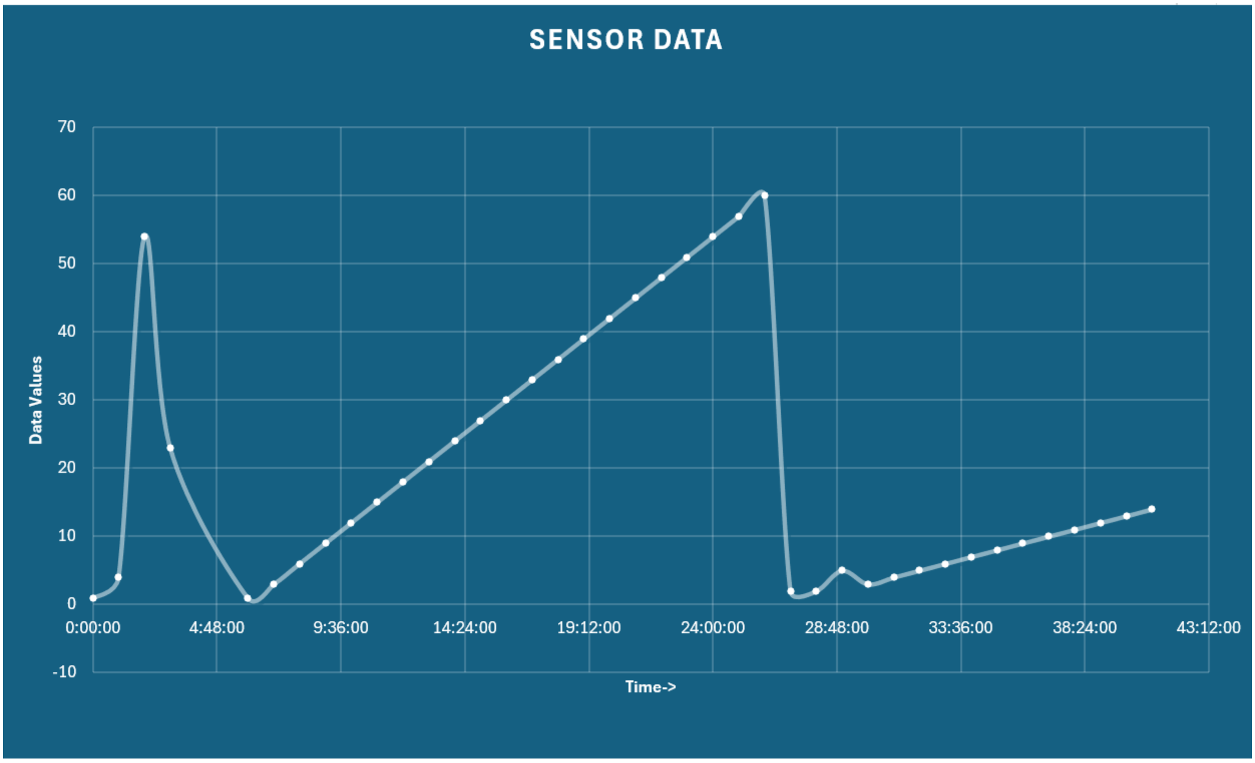

Figure 2: Sensor Data Plot

In a traditional implementation of storing data to the disk, all the data values and time (Vn, Tn) are stored to the disk using files or databases. Now let's get an idea of how much data storage is needed to store this data.

- We have 40 seconds of data.
- Assuming data is sampled every 1 millisecond or 1000 samples per second.
- Each data value takes 8 bytes of memory and corresponding time takes 8 bytes of memory too.
- This means that for storing just 40 seconds of data, we require 320,000(40 *1000 * 8) bytes for data values and 320,000(40 *1000 * 8) bytes for time.

640,000 bytes is the case for just 40 seconds from one sensor. The size required for the data storage can grow unbelievably huge as the data retention period is like months or years and may multiply with multiple data sources. This is the huge problem in almost every industry that uses computer applications which store and use the data heavily. Often companies have to compromise on reducing the data retention period or use some techniques like compression and cold storage. But these techniques come with so many limitations. Data Storage using Mathematical Model provides an extremely efficient technique to reduce storage needed for storing the data for a fairly predictable system.

Data Storage Using Mathematical Model Technique

This technique uses the idea that if a set of data can be represented by the mathematical model or an equation, then all those data points that need to be physically stored can be replaced by the mathematical model. This is by writing the start time of data, end time of the data, sampling interval and mathematical model to the datasteam and removing the actual data points(value and time).

The theory behind converting data streams to mathematical models is that any continuous data with respect to time can be converted to mathematical models. These models can be simple or complex. The state variable changes infinitely for a continuous time model within a finite time and can be represented by the equation x' = f(x, u, t) where state variable x changes over time. Often in digital representation and measurements, a discrete model can be derived by discretized version of continuous model. This is nowadays very commonly found in the IOT and Big Data world.

Let's go to the implementation part of this by considering the data from the chart above (Refer Figure 3).

Time	Data Value	Model Ref	Mathematical Model Data Stream
0:00:00	1	NA	${S:0:0:0}
1:00:00	4	NA	${I:1:0:0}
2:00:00	54	NA	1
3:00:00	23	NA	4
6:00:00	1	NA	54
7:00:00	3	1	23
8:00:00	6	2	1
9:00:00	9	3	${7:00:00}
10:00:00	12	4	${26:00:00}
11:00:00	15	5	${1:00:00}
12:00:00	18	6	${y=3x}
13:00:00	21	7	2
14:00:00	24	8	2
15:00:00	27	9	5
16:00:00	30	10	${30:00:00}
17:00:00	33	11	${41:00:00}
18:00:00	36	12	${1:00:00}
19:00:00	39	13	${y=x+2}
20:00:00	42	14	
21:00:00	45	15	
22:00:00	48	16	
23:00:00	51	17	
24:00:00	54	18	
25:00:00	57	19	
26:00:00	60	20	
27:00:00	2	NA	
28:00:00	2	NA	
29:00:00	5	NA	
30:00:00	3	1	
31:00:00	4	2	
32:00:00	5	3	
33:00:00	6	4	
34:00:00	7	5	
35:00:00	8	6	
36:00:00	9	7	
37:00:00	10	8	
38:00:00	11	9	
39:00:00	12	10	
40:00:00	13	11	
41:00:00	14	12	

Figure 3: Sensor Data and Mathematical Model Stream

The implementation strategy is to convert data points to Mathematical Model Steam.For this we need a data processing unit.Data Storage Using Mathematical Model technique uses a high speed data processing unit to convert data stream to Mathematical Model Stream however as it is a software component, this is not expensive as the physical hardware and can be maintained easily.

Following Algorithm describes the conversion of data stream to Mathematical Model Data Stream:

Step 1: First step is to write the data point start time and Interval to the mathematical model datastream as per the format below.

${S:0:0:0} ⇒ Absolute start time of the data indicated by ${<Start time>}

${I:1:0:0}⇒ Time interval of the data indicated by ${<Time Interval>}

Step 2: Buffer set of data where length of the set can be proved efficient if that much data is replaced by the model.

Step 3: Derive the mathematical model from the full set or subset within the buffered data.

Step 4: If the model is within the complexity limit and can replace a threshold number of datapoints, store the mathematical data model to the stream instead of the actual data points. Use the format below to do this.

${7:00:00} ⇒ Start time of the data indicated by ${<Start time>}

${26:00:00} ⇒ End time of the data indicated by ${<End time>}

${1:00:00} ⇒ Time interval of the data indicated by ${<Time Interval>}

${y=3x} ⇒ Actual mathematical data model indicated by ${<Mathematical Model>}

Step 5: Check if the subsequent data points satisfy the model if the model is used to derive the data till the end of the current dataset. If yes, include those points to the model.

Step 6: Go back to step 2 if more data exist to process, else go to next step

Step 7: Finish processing

Note that for a fairly predictable system, the implementation for finding the mathematical model from the buffered data can be done using a set of predefined models configured in a lookup table.

Conclusion:

Data Storage Using Mathematical Model provides an efficient way to store data in a much lesser space than the traditional way of storing the data. The high speed data processing component can be optimized to meet the memory and product requirements. This can also help in fast data transfer.

References:

1. Cellier, F.E. and Greifeneder, J., 2013. Continuous system modeling. Springer Science & Business Media.