***Software Engineering***

***By Damric Dobric/Andreas Pech***

*Migration of video learning project*

**Mashnunul Huq Nusrat Jahan Sumi**

1384042 1345476

*Abstract*: This paper represents an improving machining learning algorithm Hierarchical Temporal Memory (HTM) which is using Spatial pooler for learning Video data. The SP model shows how neurons learn by feedforward connections and form effective classification of the input frame. It converts binary input pattern into space distributed representation (SDR) by using Cortical Learning rules and homeostatic plasticity control for frame pattern prediction. The result of the learning is tested by giving the trained model an arbitrary image, the model then tries to recreate a video with proceeding frame after the input frame.

Keywords—homeostatic plasticity controller, formatting, division into frames, prediction, training & testing

# **Introduction**

The HTM (Hierarchical Temporal Memory) is based on “Thousand Brains Theory” which explains how an object behaviors and high-level concepts gets tightly replicated across a cortical column but not only on the top layer and gets distributed throughout the neocortex. Here spatial pooler involves different computational principles of the cortex. It depends on competitive Hebbian learning, homeostatic excitability control, topology of connections in sensory cortices and structural plasticity. The HTM Spatial pooler is developed in such a way to achieve a set of computational properties which includes 1. Preserving topology of the input space by mapping similar inputs to similar outputs 2. Continuously adapting to changing statistics of the input stream 3. Forming fixed sparsity representations 4. Being robust to noise and 5. Being fault tolerant that supports computations with SDRs (Sparse Distributed Representations). The output of the SP which is the integral component of HTM can be easily recognized by downstream neurons and contribute to improved performance in the end-to-end HTM system.

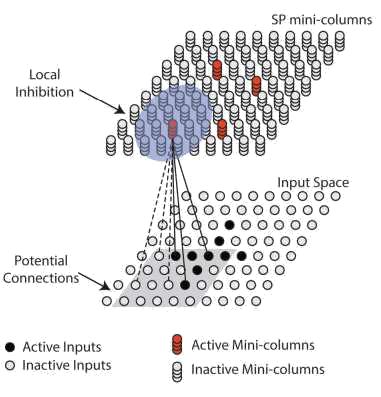


Figure 1. The process of spatial pooling

SP plays a vital role in HTM networks. The task of SP is to transforms input patterns into Sparse Distributed Representation in a continuous way in end-to-end HTM system. The temporal sequences of these SDRs is learned by the HTM and do some prediction for the upcoming inputs. A single layer in HTM network is consist of a set of mini-columns which is consist of cells. Here the figure 1. shows that the HTM spatial pooler converts inputs at bottom to SDRs at top. Each SP mini-column (Active mini-columns and inactive Mini-columns) forms synaptic connections to a subset of the input space which is consist of gray square and potential connections. A local inhibition technique gives confirmation that within the local inhibition radius (shaded blue circle) a small fraction of the SP mini-columns that receive most of the inputs are active. According to the Hebbian rule Synaptic permanences are adjusted like this for each active SP mini-column, active inputs (black lines) are reinforced and inactive inputs (dashed lines)are punished.

At the time of building intelligent systems to mimic human intelligence and cognition, we must pay serious attention to sequences, including sequence learning as sequential behavior is essential to intelligence. In the task of time series prediction, video analysis and musical information retrieval, a model must learn from inputs that are sequences. Another important concern in sequence learning is hierarchical structuring of sequences. Many real-world problems that have sequences are involved with clear hierarchical structures like a sequence is made up of subsequences and they in turn are made up of sub subsequences and so on. By removing the difficulty to identify automatically these subsequences and deal with them accordingly which is related to temporal dependences, learning hierarchical structures help to reduce or eliminate temporal dependencies. It helps to compress the description or sequences.

Sequence learning is not a easy task. Sequence learning are needed powerful algorithms. Sequence learning which indicates either generation, prediction or recognition is usually based on the models of legitimate sequences which can be developed through training with exemplars. Hierarchical Temporal Memory proposed new computational learning models, Cortical Learning Algorithms (CLA), that is inspired from the neocortex which offer a better understanding of how our brains function. CLA mimics the procedure of human brain how to achieve pattern recognition and make intelligent predictions. The CLA processes the streams of information, classify them, learning to identify the differences and using time-based patterns to make predictions as like as performed by the neocortex in humans. But the place of time is significant in case of learning, inference and prediction. The temporal sequence is achieved from HTM algorithm from the stream of input data. Here Afterwards the result of the learning is tested by giving the trained model an arbitrary image, the model then attempts to recreate a video with proceeding frame after the input frame.

This type of works forecasting that Machine Learning (ML) or statistical modelling emphasis here is to enable the reader to understand on some of ML or statistical techniques actively used in past and till the present moment.

# Mthods

## Creating Video Data Files

There are different ways to create training videos of object recognition but we chose to create our object videos using python OpenCV library. As we worked on the previous “[**neocortexapi-videolearning**](https://github.com/ddobric/neocortexapi-videolearning)” project we had video data set for recognizing circle, triangle and rectangle. With the help of previous python [codes](https://github.com/ddobric/neocortexapi-videolearning/tree/main/DataGeneration) we created training video-set for a [line](https://github.com/ddobric/neocortexapi-videolearning/blob/videolearning-migration/DataGeneration/line.py) moving around the 120x120 frame with a frame rate of 24 frames per second and the thickness of the object is 8. Videos can be found “SmallTrainingSet” folder.

## Reading Video

To train a video to a machine learning program it has to be divided it into picture frames. As like brain, frames are the point of reference to the recognition program. As we are predicting future frames of an object or in plain language next move of the object according to it’s behavior we need to give as small distinctive data as possible to the program for reducing computational time. By [VideoSet](https://github.com/ddobric/neocortexapi-videolearning/blob/videolearning-migration/VideoLibrary/VideoSet.cs) class from the supporting library of the project [VideoLibrary](https://github.com/ddobric/neocortexapi-videolearning/tree/videolearning-migration/VideoLibrary) we extract frames of each video.

As the previous videos of this training data set had the same frame rate as our line data the video configuration is not changed. We implemented a [videoConfig.json](https://github.com/ddobric/neocortexapi-videolearning/blob/videolearning-migration/HTMVideoLearning/videoConfig.json) file for the ease of configuration changes to the videos implemented in the future for training. The training video rate is reduced to half of the original video frame rate(see Table 2)for making more frames and for computational ease of data to be introduced in the system the frame size is also reduced to 18x18 pixels.

|  |
| --- |
| frameWidth": 18,  frameHeight": 18,  frameRate: 12,  ColorMode: "BLACKWHITE" |

Table : Video Configuration

## Converting Frames to bitarrays:

For learning in the spatial pooler and temporal memory each frame has to be binarized. By [NFrame](https://github.com/ddobric/neocortexapi-videolearning/blob/videolearning-migration/VideoLibrary/NFrame.cs) class we binarized each frame into binary array using BitmapToBinaryArray method. (see Table 3) We used Black&White format to binarize each frame as our training videos are created based on Black and White color mode for ease of computational time required for processors. Pure or RGB color mode requires great amount of computational time as per our tests. When the predicted images are recreated IntArrayToBitmap method is used.

|  |
| --- |
| private int[] BitmapToBinaryArray(Bitmap image)  {  for (int heightCount = 0; heightCount < img.Height; heightCount++)  {  for (int widthCount = 0; widthCount < img.Width; widthCount++)  {  switch (colorMode)  { imageBinary.Add((luminance > 255 / 2) ? 0 : 1);  imageBinary.AddRange(new List<int>() { (pixel.R > 255 / 2) ? 1 : 0, (pixel.G > 255 / 2) ? 1 : 0, (pixel.B > 255 / 2) ? 1 : 0 });  imageBinary.AddRange(ColorChannelToBinList(pixel.R));  imageBinary.AddRange(ColorChannelToBinList(pixel.G));  imageBinary.AddRange(ColorChannelToBinList(pixel.B));  }  }  }  } |

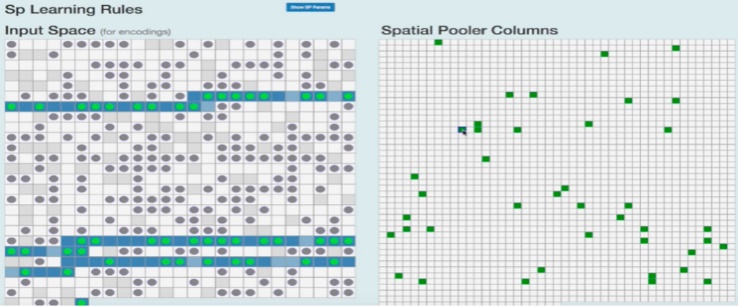
Table : Frame to Bitarray

## Learning Stage

Now the binarized frames are sent to Spatial Pooler which operates on mini-columns to sensory inputs (for this case the movement of the object, it’s thickness, size, direction etc.) and learn spatial patterns by encoding the pattern into Sparse Distributed Representation (SDR)



|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | … | 1 | 1 | 1 | 0 | … | 1 | 1 | 1 |
| 1 | 1 | 0 | … | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | … | 1 | 1 | 1 | 1 | 1 | 1 | 0 | … |
| 1 | 1 | 1 | 1 | 1 | 0 | … | 1 | 1 | 1 |
| 1 | 0 | … |  |  |  |  |  |  |  |



The created SDR which is the encoded spatial pattern of that object is used as the input to the Temporal Memory which learns about the patter when the spatial pooler is instable mode and removes the pattern when it is in unstable mode. SP oscillates between stable and unstable mode and the TM also learns and forgets about the pattern. But too much oscillation can cause permanent disruption to the program hence causing higher computational resources. To reduce this scenario we used homeostatic plasticity controller which influences excitation and inhibition balance of neurons. The functional stability of neural columns is achieved by SP and TM setting cells in active or predictive state. SP provides Global and Local inhibition which controls the number of cells must be activated in the currently processing area. To keep the stability of the Spatial Pooler and learning of TM a set of common parameters were selected while instantiating HTM () and kept in the [htmConfig.json](https://github.com/ddobric/neocortexapi-videolearning/blob/videolearning-migration/HTMVideoLearning/htmConfig.json) file. Some of the configurations are manipulated while running the program in ModifyHtmFromCode method in the Main [Program](https://github.com/ddobric/neocortexapi-videolearning/blob/videolearning-migration/HTMVideoLearning/Program.cs) class.

|  |  |
| --- | --- |
| Parameters | Value |
| CellsPerColumn | 30 |
| GlobalInhibition | true |
| NumActiveColumnsPerInhArea | 0.02\*ColumnDimension |
| PotentialRadius | 0.15\* InputDimension |
| MaxBoost | 30.0 |
| MaxSynapsesPerSegment | 0.02\*ColumnDimension |
| Random | 42 |
| MinPctOverlapDutyCycles | 0.75 |
| DutyCyclePeriod | 100 |
| StimulusThreshold | 0.05\*ColumnDimension |
| UpdatePeriod | 50 |
| PermanenceIncrement | 0.1 |
| PermanenceDecrement | 0.01 |

Now the boosting in spatial pooler makes sure that all columns are uniformly used across all seen patterns. As the mechanism remains active throughout the process the boosting of columns which already build learned SDRs is possible. Deactivation of boosting in homeostatic plasticity in the cortical layer can also be applied to SP. But the actual understanding to this is yet to be revealed. Till now in HTM this technique consists of boosting and inhibition algorithms which works on the minimum column level and not on the cell level in the minimum column. Because SP operates on the population of neural cells in minimum column rather than the individual cells(). Therefore, the Spatial Pooler with the

New-born Stage is used with the aim to send input pattern of SDR in each iteration to the homeostatic plasticity controller telling the program that SP has reached instable stage and program will disable the boosting. As the SP has entered to a stable state it will leave the new-born cycle and continue operating as usual without boosting which will help in reducing computational time.

For differentiating multi sequence learning and sequence learning we instantiated HtmClassifier with two different approaches. In the sequence learning method defined in [VideoLearning](https://github.com/ddobric/neocortexapi-videolearning/blob/videolearning-migration/HTMVideoLearning/VideoLearning.cs) class as TrainWithFrameKey, we put the frame key as HtmClassifier key while calling for the learn method. On the other hand for sequential learning we used series of frame as the HtmClassifier key while calling for the learn method (). By the definition sequence learning should take more computational time while learning as it learns by each frame. But the multi sequence learning should take less time as it takes a bunch of frames while learning.

|  |
| --- |
| public static void TrainWithFrameKey(VideoConfig videoConfig = null, HtmConfig htmCfg = null)  {  HtmClassifier<string, ComputeCycle> cls = new();  HomeostaticPlasticityController hpa = new(mem, maxNumOfElementsInSequence \* 150 \* 3, (isStable, numPatterns, actColAvg, seenInputs) =>{}, numOfCyclesToWaitOnChange: 50)  SpatialPoolerMT sp = new(hpa);  for (int i = 0; i < maxCycles; i++)  {  foreach (var currentFrame in nv.nFrames)  {  cls.Learn(currentFrame.FrameKey, actCells.ToArray());  cls.Learn(key, actCells.ToArray()); //For TrainWithFrameKeys  }  }  } |

## Predicting Frames from an input

After the learning we counted the accuracy of each learned video by calling a method PredictImageInput of [VideoLearning](https://github.com/ddobric/neocortexapi-videolearning/blob/videolearning-migration/HTMVideoLearning/VideoLearning.cs) class which takes an image and recreates the consecutive frames after that image. This is done in two stages. First from image directories given in the videoConfig.json file () and then taking images as directory path from users. The directory holding testing videos without input from user contains frames created by the program as it is required to test from the given input also.

As all the data is binarized, these images also needs to be binarized with [NFrame](https://github.com/ddobric/neocortexapi-videolearning/blob/videolearning-migration/VideoLibrary/NFrame.cs) class’s BitmapToBinaryArray method. While making the predicted future frames IntArrayToBitmap method of the same class is used and then combining all of these frames are done in [NVideo](https://github.com/ddobric/neocortexapi-videolearning/blob/videolearning-migration/VideoLibrary/NVideo.cs) class’s CreateVideoFromFrames method and saves those in a directory called convertedVideoDir. As the program was built on old version [Emgu.cv](https://www.emgu.com/wiki/files/4.1.0/document/html/3fb90645-ecc4-0c4e-b238-6d0ca38f4ebc.htm) framework, this method used to have -1 called while initiating VideoWriter object for the manual selection of coding-decoding format of the video which is now obsolete. Now user can select the video format while calling CreateVideoFromFrames method as we introduced fourcc for format selection(see Table 1)but the default is mp4 format.

We also have calculated the accuracy on the training dataset as well as the testing data set. The accuracy is calculated using equation:

 



The accuracy reaches to saturation and after getting 10 similar accuracy the program moves to next cycle to reduce computational time(). If the accuracy is more than 80% then it is recorded with the predicted video.

|  |
| --- |
| "TestFiles":[  "Run2ExperimentOutput\\Converted\\Circle\\circle\\Circle\_circle\_3.png",  "Run2ExperimentOutput\\Converted\\Circle\\circle\\Circle\_circle\_2.png",  "Run2ExperimentOutput\\Converted\\Line\\line\\Line\_line\_11.png”, "Run2ExperimentOutput\\Converted\\Line\\line\\Line\_line\_22.png", "Run2ExperimentOutput\\Converted\\Rectangle\\rectangle\\  Rectangle\_rectangle\_28.png", "Run2ExperimentOutput\\Converted\\Rectangle\\rectangle\\  Rectangle\_rectangle\_18.png",  "Run2ExperimentOutput\\Converted\\Triangle\\triangle\\  Triangle\_triangle\_23.png", "Run2ExperimentOutput\\Converted\\Triangle\\triangle\\  Triangle\_triangle\_0.png",  ] |

|  |
| --- |
| public static void CreateVideoFromFrames(List<NFrame> bitmapList, string videoOutputPath, int frameRate, Size dimension, bool isColor, char[] codec = null )  {  int fourcc = VideoWriter.Fourcc(codec[0], codec[1], codec[2], codec[3]);  using (VideoWriter videoWriter = new("videoOutputPath..mp4", fourcc, (int)frameRate, dimension, isColor))  } |

|  |
| --- |
| // Accuracy Check  double accuracy;  accuracy = (double)matches / ((double)nv.nFrames.Count - 1.0) \* 100.0;  if (accuracy == lastCycleAccuracy)  {  saturatedAccuracyCount += 1;  if (saturatedAccuracyCount >= 10 && lastCycleAccuracy >= 80)  {  }  } |

# Prepare Your Paper Before Styling

Before you begin to format your paper, first write and save the content as a separate text file. Complete all content and organizational editing before formatting. Please note sections A-D below for more information on proofreading, spelling and grammar.

Keep your text and graphic files separate until after the text has been formatted and styled. Do not use hard tabs, and limit use of hard returns to only one return at the end of a paragraph. Do not add any kind of pagination anywhere in the paper. Do not number text heads-the template will do that for you.

## Abbreviations and Acronyms

Define abbreviations and acronyms the first time they are used in the text, even after they have been defined in the abstract. Abbreviations such as IEEE, SI, MKS, CGS, sc, dc, and rms do not have to be defined. Do not use abbreviations in the title or heads unless they are unavoidable.

## Units

* Use either SI (MKS) or CGS as primary units. (SI units are encouraged.) English units may be used as secondary units (in parentheses). An exception would be the use of English units as identifiers in trade, such as “3.5-inch disk drive”.
* Avoid combining SI and CGS units, such as current in amperes and magnetic field in oersteds. This often leads to confusion because equations do not balance dimensionally. If you must use mixed units, clearly state the units for each quantity that you use in an equation.
* Do not mix complete spellings and abbreviations of units: “Wb/m2” or “webers per square meter”, not “webers/m2”. Spell out units when they appear in text: “. . . a few henries”, not “. . . a few H”.

may use the solidus ( / ), the exp function, or appropriate exponents. Italicize Roman symbols for quantities and variables, but not Greek symbols. Use a long dash rather than a hyphen for a minus sign. Punctuate equations with commas or periods when they are part of a sentence, as in:

*a**b* 

Note that the equation is centered using a center tab stop. Be sure that the symbols in your equation have been defined before or immediately following the equation. Use “(1)”, not “Eq. (1)” or “equation (1)”, except at the beginning of a sentence: “Equation (1) is . . .”

## Some Common Mistakes

* The word “data” is plural, not singular.
* The subscript for the permeability of vacuum **0, and other common scientific constants, is zero with subscript formatting, not a lowercase letter “o”.
* In American English, commas, semicolons, periods, question and exclamation marks are located within quotation marks only when a complete thought or name is cited, such as a title or full quotation. When quotation marks are used, instead of a bold or italic typeface, to highlight a word or phrase, punctuation should appear outside of the quotation marks. A parenthetical phrase or statement at the end of a sentence is punctuated outside of the closing parenthesis (like this). (A parenthetical sentence is punctuated within the parentheses.)
* A graph within a graph is an “inset”, not an “insert”. The word alternatively is preferred to the word “alternately” (unless you really mean something that alternates).
* Do not use the word “essentially” to mean “approximately” or “effectively”.
* In your paper title, if the words “that uses” can accurately replace the word “using”, capitalize the “u”; if not, keep using lower-cased.
* Be aware of the different meanings of the homophones “affect” and “effect”, “complement” and “compliment”, “discreet” and “discrete”, “principal” and “principle”.
* Do not confuse “imply” and “infer”.
* The prefix “non” is not a word; it should be joined to the word it modifies, usually without a hyphen.
* There is no period after the “et” in the Latin abbreviation “et al.”.
* The abbreviation “i.e.” means “that is”, and the abbreviation “e.g.” means “for example”.

An excellent style manual for science writers is [7].

# Using the Template

After the text edit has been completed, the paper is ready for the template. Duplicate the template file by using the Save As command, and use the naming convention prescribed by your conference for the name of your paper. In this newly created file, highlight all of the contents and import your prepared text file. You are now ready to style your paper; use the scroll down window on the left of the MS Word Formatting toolbar.

## Authors and Affiliations

**The template is designed for, but not limited to, three authors.** A minimum of one author is required for all report articles. Author names should be listed starting from left to right and then moving down to the next line. This is the author sequence that will be used in future citations and by indexing services. Names should not be listed in columns nor group by affiliation. Please keep your affiliations as succinct as possible (for example, do not differentiate among departments of the same organization).

### For papers with more than three authors: Add author names horizontally, moving to a third row if needed for more than 8 authors.

### For papers with less than three authors: To change the default, adjust the template as follows.

#### Selection: Highlight all author and affiliation lines.

#### Change number of columns: Select the Columns icon from the MS Word Standard toolbar and then select the correct number of columns from the selection palette.

#### Deletion: Delete the author and affiliation lines for the extra authors.

## Identify the Headings

Headings, or heads, are organizational devices that guide the reader through your paper. There are two types: component heads and text heads.

Component heads identify the different components of your paper and are not topically subordinate to each other. Examples include Acknowledgments and References and, for these, the correct style to use is “Heading 5”. Use “figure caption” for your Figure captions, and “table head” for your table title. Run-in heads, such as “Abstract”, will require you to apply a style (in this case, italic) in addition to the style provided by the drop down menu to differentiate the head from the text.

Text heads organize the topics on a relational, hierarchical basis. For example, the paper title is the primary text head because all subsequent material relates and elaborates on this one topic. If there are two or more sub-topics, the next level head (uppercase Roman numerals) should be used and, conversely, if there are not at least two sub-topics, then no subheads should be introduced. Styles named “Heading 1”, “Heading 2”, “Heading 3”, and “Heading 4” are prescribed.

## Figures and Tables

#### Positioning Figures and Tables: Place figures and tables at the top and bottom of columns. Avoid placing them in the middle of columns. Large figures and tables may span across both columns. Figure captions should be below the figures; table heads should appear above the tables. Insert figures and tables after they are cited in the text. Use the abbreviation “Fig. 1”, even at the beginning of a sentence.

1. Table Type Styles

| Table Head | Table Column Head | | |
| --- | --- | --- | --- |
| Table column subhead | Subhead | Subhead |
| copy | More table copya |  |  |

1. Sample of a Table footnote. (*Table footnote*)
2. Example of a figure caption. (*figure caption*)

Figure Labels: Use 8 point Times New Roman for Figure labels. Use words rather than symbols or abbreviations when writing Figure axis labels to avoid confusing the reader. As an example, write the quantity “Magnetization”, or “Magnetization, M”, not just “M”. If including units in the label, present them within parentheses. Do not label axes only with units. In the example, write “Magnetization (A/m)” or “Magnetization {A[m(1)]}”, not just “A/m”. Do not label axes with a ratio of quantities and units. For example, write “Temperature (K)”, not “Temperature/K”.

##### Acknowledgment *(Heading 5)*

The preferred spelling of the word “acknowledgment” in America is without an “e” after the “g”. Avoid the stilted expression “one of us (R. B. G.) thanks ...”. Instead, try “R. B. G. thanks...”. Put sponsor acknowledgments in the unnumbered footnote on the first page.

##### References

The template will number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2]. Refer simply to the reference number, as in [3]—do not use “Ref. [3]” or “reference [3]” except at the beginning of a sentence: “Reference [3] was the first ...”

Number footnotes separately in superscripts. Place the actual footnote at the bottom of the column in which it was cited. Do not put footnotes in the abstract or reference list. Use letters for table footnotes.

Unless there are six authors or more give all authors’ names; do not use “et al.”. Papers that have not been published, even if they have been submitted for publication, should be cited as “unpublished” [4]. Papers that have been accepted for publication should be cited as “in press” [5]. Capitalize only the first word in a paper title, except for proper nouns and element symbols.

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [6].

1. G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. *(references)*
2. J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.
3. I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.
4. K. Elissa, “Title of paper if known,” unpublished.
5. R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.
6. Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].
7. M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.

**This report template contains guidance text for composing and formatting technical reports. Please ensure that all template text is removed from your report prior to submission to the examination office. Failure to remove template text from your paper may result in your paper being degraded.**