# Realizing Forgetting in a Modified Sparse Distributed Memory System

**Uma Ramamurthy (urmmrthy@memphis.edu)**
**Sidney K. D'Mello (sdmello@memphis.edu)**
**Stan Franklin (franklin@memphis.edu)**

Computer Science Department & The Institute for Intelligent Systems,
The University of Memphis,
365 Innovation Drive, Memphis, TN 38152, USA

## Abstract

This paper presents research on the development of effective forgetting mechanisms for the Sparse Distributed Memory (SDM) system, to computationally model Transient Episodic Memory (TEM), a short-term sensory perceptual episodic memory in software agents. Possible theories and mechanisms for forgetting are *retrieval failures, decay* and *interference*. The SDM architecture has inherent features to effect interference and retrieval failures. We have implemented two decay mechanisms in a variant of the SDM system. In this paper, we present the decay mechanisms and the experimental results. The results show that the decay mechanisms compliment the inherent features of the SDM architecture in realizing forgetting for TEM.

## Introduction

It is well established that, in content-addressable, associative, episodic memories[1], interference results when similar events over time become merged into a general event, blurring their details (Chandler, 1991; Lenhart & Freeman, 1995; Lindsay & Read, 1995). Thus, declarative memory (long-term memory for autobiographical events and semantic knowledge) cannot be counted on to help with the recall of where one parked one's car in the parking garage this morning or what one had for lunch yesterday. These events are much too similar to a myriad of earlier such events. Yet such recall is essential for cognitive functioning. One needs to know where to find one's car.

In order to circumvent these functional difficulties associated with the retrieval of detailed information of recent events we hypothesize that humans have a content-addressable, associative, transient episodic memory (TEM) with an information retention period measured in hours (Baars & Franklin, 2003; Franklin, Baars, Ramamurthy & Ventura, 2005). Humans are able to recall in great detail events of the immediate past – where they park their cars, whom they met that morning, what they discussed, what they had for meals, etc. The details of these events/episodes stay with us only for short durations – a few hours to a day. For different empirical reasons, Conway postulates a sensory-perceptual episodic memory (similar to TEM) with an information retention period measured in hours or perhaps a day, and with a sizable capacity (2001). Donald also assumes a similar TEM which he calls an intermediate-term working memory (2001), while Panksepp speaks of a "transient memory store" (1998, page 129). Baddeley has proposed that working memory includes an episodic buffer that can hold episodic information for a short duration (2000).

In order to achieve an acceptable degree of specificity as required by TEM, an effective mechanism for forgetting needs to be in place. Two primary theories and possible mechanisms of forgetting are *decay* (Brown, 1958; Ebbinghaus, 1985/1964; Peterson & Peterson, 1959) and *interference* (Keppel & Underwood, 1962; McGeoch, 1932; Waugh & Norman, 1965). Interference influences forgetting because similar events encoded in a memory system interfere with one another and negatively affect retrieval. Alternately, decay brings about forgetting by causing a loss of memory traces attributed only to time. *Retrieval failures* have also been proposed as the possible basis for forgetting – memories never disappear; they just cannot be retrieved (Tulving, 1968).

Our interest with human memory systems emerges from our desire to model several faculties of human (and animal) cognition by developing cognitive agents (software and robotic) capable of robust autonomy. The Intelligent Distribution Agent (IDA) is a cognitive software agent (Franklin, 1997; 2001) developed for the U.S. Navy. At the end of each sailor's tour of duty, he or she is assigned to a new billet by a person called a *detailer*. IDA's task is to facilitate this process by completely automating the role of a detailer. The design of the IDA technology and its more recent learning extension (LIDA) is motivated by a number of new AI techniques. The IDA architecture has a number of different memory systems, including sensory memory, perceptual associative memory, working memory, transient episodic memory, procedural memory, and declarative (autobiographical + semantic) memory.

Transient episodic and declarative memories have distributed representations in IDA. There is evidence that this is also the case in animal nervous systems. The memory systems are computationally modeled by Sparse Distributed Memory (Kanerva, 1988). This is reasonable due to several functional and neural similarities between SDM and human memory systems. The functional parallels include SDM's ability to account for classical memory phenomena such as *knowing that one knows, the tip-of-the-tongue effect, rehearsal, momentary feelings of familiarity,* and *interference.* The neural similarities between SDM and

[1] In *content addressable* memories retrieval of a stored pattern is based on its degree of similarity to a retrieval cue, and not to an explicit address like a computer memory (RAM). An *associative* memory makes associations between related patterns, such that when one is encountered, the related patterns can be recalled. *Episodic* memories encode events with semantic, spatial, and temporal features, i.e., the *what*, the *where*, and the *when*.

human memory emerge from the likeness of the mathematical formulation of SDM to models of the cerebellar cortex developed by Marr (1969) and Albus (1971) (Kanerva, 1993).

The focus of this paper is on the development of effective forgetting mechanisms for a variant of the SDM architecture, the modified SDM system (Ramamurthy, D'Mello, & Franklin, 2004), which shows promise to be a good candidate for use as a TEM in software agents such as IDA.

## Theoretical Background

### Sparse Distributed Memory

SDM implements a content-addressable random access memory. Its address space is in the order of $2^d$ (*d* is the dimensionality of the space and the size of the patterns). Of this space, you choose a manageable, uniform random sample, say *m*, of allowable locations. These are called *hard locations*. Thus the hard locations are sparse in this address space. Many hard locations participate in storing and retrieving of any datum, resulting in the distributed nature of this architecture. Hamming distance is used to measure the distance between any two points in this memory space.

Each hard location is a bit vector of length *d*, storing data in *d* counters with a predefined limit. Each datum to be written to SDM is a bit vector of length *d*. Writing 1 to a counter results in incrementing the counter, while writing a 0 decrements the counter. To write in this memory architecture, you select an access sphere centered at location X. To write a datum to X, you simply write to all the hard locations (typically *1%* of *m*) within X's access sphere. This results in distributed storage. This also naturally provides for memory rehearsal – a memory trace being rehearsed can be written many times and each time to about *0.01 times m* locations.

Similar to writing, retrieving from SDM involves the same concept of access sphere – you read all the hard locations within the access sphere of location Y, pool the bit vectors read from all these hard locations and let each of the $k^{th}$ bits of those locations participate in a majority vote for the $k^{th}$ bit of Y. You reconstruct the memory trace in every retrieval operation. Effectively, the read data at Y is an aggregate of all data that have been written to the hard locations within Y's access sphere, but may not be any of them exactly.

Furthermore, this memory can be cued with noisy versions of the original memory trace. To accomplish this, you employ iterated reading – first read at Y to obtain the bit vector, Y1. Next read at Y1 to obtain the bit vector Y2. Next read at Y2 to obtain the bit vector, Y3. If this sequence of reads converges to Y', then Y' is the result of iterated reading at Y. Please see Kanerva (1988) for details.

### The Modified SDM system

A preliminary experimental evaluation of Kanerva's original SDM for cognitive agents such as IDA, that encode text based episodic data, indicated the need for an architecture modification. Episodic data refers to patterns with features of the what, the where, and the when. When events are unfolding, the feature vector (the pattern written to memory) is not always complete. So, more often, the agent has to store partial feature sets. Similarly, when the agent cues its memory for retrieval, the retrieval cues are often partial feature-sets. SDM has no generic mechanism to handle partiality in the stored patterns as well as in the retrieval cues. It considers missing features to be random noise, thereby severely effecting performance.

The modified SDM system (Ramamurthy, D'Mello, & Franklin, 2004) alleviates this problem of encoding and retrieval with partial patterns. The modification includes migrating to a ternary memory space while maintaining a binary address space for the hard locations. Adding "don't cares" (*'s) to the 0's and 1's of the binary space yields a ternary memory space. This accommodates flexible cuing with fewer features than the actual memory trace where missing features are represented by "don't cares" (*). Additionally, an adjustment was made to the Hamming distance calculations such that the distance between a "don't care" (*) and a 0 or 1 was set to (0.5).

Detailed experimental simulations on the modified SDM system show a significant improvement in performance when compared to the original SDM system (D'Mello, Ramamurthy, & Franklin, 2005; Ramamurthy, D'Mello, & Franklin, 2004). The modified SDM system demonstrated more efficient distribution of the encoded patterns across the hard locations in the memory space. Its abilities in encoding partial patterns and retrieval with partial cues are also significantly better than the original SDM. Interestingly, a reasonable degree of "don't cares" in the stored patterns improves performance as they act as attractor basins due to the modification to the Hamming distance calculation. Additionally, the modified SDM system also alleviates some of the problems related to text encoding (see below) by its improved retrieval quality when compared to the original SDM system. However, without appropriate forgetting mechanisms in place, we suspect that the modified SDM system will be unable to deliver the desired retrieval accuracies as required by TEM.

### Rationale for Decay in the Modified SDM system

Historically decay and interference have been proposed as two theories of forgetting. It would clearly be beneficial if we could rely on interference as the exclusive mechanism of forgetting in TEM. Due to SDM's massively distributed architecture, where each pattern is encoded to approximately one hundredth of the hard locations, forgetting due to interference is a bi-product of the system. However, while experimental simulations have verified the effect of interference in SDM, in certain situations, the degree to which encoded patterns interfere with each other can have adverse effects.

A potential cause for undue information corruption due to undesirable interference effects emerges from SDM's poor performance in encoding patterns consisting of non-random data. D'Mello, Ramamurthy, and Franklin have reported results of simulations where even when the memory was filled to capacity, with text-based episodic data, only

33.05% and 25.01% of the hard locations in the modified and original SDM respectively were involved in the encoding process (2005). This implies a clustering of the patterns in about a third of the memory space which would potentially cause undesirable interference effects. These results are consistent with the notion of SDM's performance failures for handling non-random data (Hely, Willshaw & Hayes, 1997) and in some sense are a justification for a domain based initialization approach (Fan & Wang, 1997; Rao & Ballard, 1995) as opposed to the conventional random initialization utilized in these experiments.

The undesirable interference effects caused by poor distribution of non-random data are amplified when text-based information is encoded into SDM. Since SDM operates in a Boolean space, encoding text requires a binary representation of the characters. A simple way to enforce this mapping is by encoding the ASCII representation of characters. For example, the feature "dog", would be represented as "01100100 01101111 01100111". Since interference from related features effects the retrieved trace, error in recall is introduced. During the recall procedure, if the second bit of each character in the binary representation of dog is flipped, the resultant binary pattern is "0*0*100100 0*0*101111 0*0*100111". Converting this recalled binary pattern into text would result in "$/'", which at the character level bears absolutely no similarity to "dog." This simple example shows that a 12.5% error in the retrieval process can completely distort the feature. It should be noted that in some cases where one or two characters in a retrieved feature are corrupted, the correct feature (word) can be retrieved by the application of approximate string matching algorithms (Baeza-Yates & Navarro, 1999; Knuth, Morris, & Pratt, 1977) that are similar to spell checkers in commercial word processors. However, we refrain from using such methods, because the use of such techniques does not seem to be cognitively plausible.

Although the modified SDM system does relax some of the adverse interference effects of the original SDM system, it does not solve the problems to an acceptable degree. Therefore, we propose the use of decay to compensate for some of the interference related problems in SDM. This approach has been considered plausible in explaining decay in short-term memory systems (Rettman, 1971). More recently Altmann and Gray have proposed a theory that functionally relates decay and interference (2002). The fundamental premise of their theory is rooted in the fact that if a memory trace decays, it causes lower interference with future memory traces. It should be noted that the notion of decay in both short-term and long-term memories is a matter of intense debate. While we use decay to alleviate specific computational problems with the modified SDM as a model of TEM, we refrain from making any controversial statements regarding the influence of decay in human memory.

## Decay Mechanisms for the Modified SDM System

There is a direct relationship between the values in the counters of the hard locations and the memory traces stored in the modified SDM. The strength of the memory traces is directly proportional to the number of times the memory traces have been rehearsed. To affect decay of stored memory traces in the modified SDM, the contents of the counters in each of the hard locations were decremented on the basis of the decay function employed. Mathematical formulations of two plausible decay mechanisms, an exponential decay function, and a negated-translated sigmoid function, are presented in Table 1.

Table 1: Mathematical Formulation of Decay Mechanisms

| Decay Mechanism | Mathematical Function |
|---|---|
| Exponential | $f(x) = 1 + e^{-ax}$ |
| Negated-Translated Sigmoid | $f(x) = 1 - \left[ \frac{1}{1 + e^{-a(x-c)}} \right]$ |

In the exponential decay function (Figure 1), the decay rate (*f*) approaches zero exponentially as the counter values (*x*) increase, without ever reaching zero. For low values of the counters, the decay rate is high and the decay rate approaches zero as the counter values increase. The decay rate drops sharply with this function. Figure 1 shows a set of exponential decay curves for different values of the parameter '*a*'.

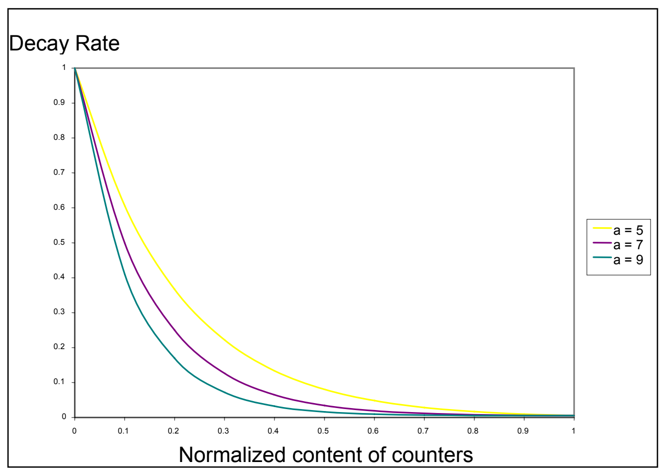


Figure 1: Exponential Decay Curves

The negated-translated sigmoid decay function shown in Figure 2 is, in principle, similar to the exponential decay function with respect to the change in decay rate. The decay rate approaches zero asymptotically as x increases, without ever reaching zero. The function is obtained by first negating the classic sigmoid function and then translating the negated function by positive 1. In contrast to the exponential decay function, this decay function has a smoother drop in the initial high decay rate. For higher values of the counters, the decay rate is closer to zero, while for low values of the counters, the decay rate is high.

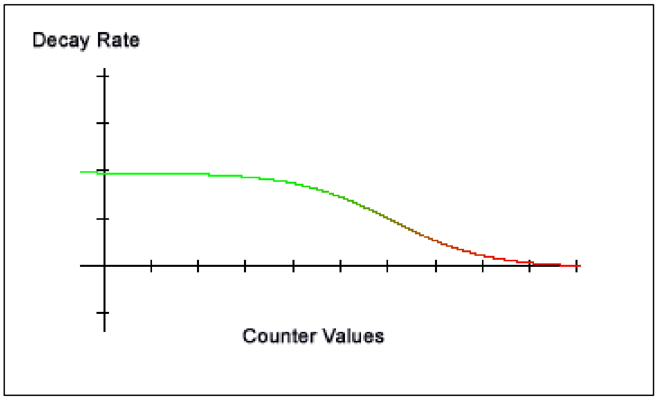


Figure 2: Negated-Translated Sigmoid Decay Curve

**TEMPLATE:**
**"Agent | Verb | Recipient-Adjective | Recipient | Object-Adjective | Object | Place | Time"**

**EXAMPLES:**
**Test-A:**
**(1) "Richard | drives | joyful | Vanessa | lively | comedy | Theatre | Friday"**
**(2) "Michael | answers | cousin | Nathan | nervous | queries | eatery | Tuesday"**
**Test-B:**
**(1) "Richard | drives | * | * | lively | comedy | * | Friday"**
**(2) " * | answers | * | Nathan | * | queries | * | Tuesday"**

Figure 3: Case-grammar template with example episodes

## Experimental Simulations

The modified SDM with these two decay mechanisms was tested with several types of memory traces. All the tests were aimed at determining the ability of TEM with decay to forget less rehearsed (written) memory traces, while preserving traces that were sufficiently reinforced. It is hypothesized that retained episodes are consolidated to the declarative memory (DM) at a later point in time. A predicted side effect of the forgetting process would be reduced interference among the encoded memory traces.

### Qualitative Review of Decay

The purpose of simulations presented here was to qualitatively assess the performance of the two decay mechanisms. All memory traces were formulated on the basis of the case-grammar template illustrated in Figure 3. Each episodic set consisted of six related episodes (high inter association) sharing the same agent and recipient, with varying spatial and temporal features.

The first set of tests (Test-A) used fully specified memory traces (no missing features) while the second set (Test-B) used partial read cues for retrieval. Of the 6 related events in each set of episodes, one event in each set was written a large number of times (well *rehearsed*) while the other 5 events in each set were encoded with smaller, varying strengths (*rehearsed less*). The effect of decay was observed by conducting several memory retrieval operations over multiple decay cycles.

For the exponential decay, the system was tested for several values of parameter *'a'*, ranging from 3 to 9. With the negated-translated sigmoid decay, the system was tested for three values of '*a*', namely, 2, 3 and 4 and value of '*c'* was set at 3. For values of '*a*' greater than 4 in the negated-translated sigmoid decay, the initial high decay rate drops almost in the same fashion as the exponential decay.

The testing of the two decay mechanisms was evaluated on the basis of the number of cycles taken for the most rehearsed/written memory trace(s) to decay away and hence the system's inability to retrieve those memory traces or *forget* the memory traces. We also considered the total number of cycles required for all the memory traces to decay.

The results indicate that the "don't cares" in the content space of the modified SDM were not a predictive factor in the decay process. Irrespective of partial writes and partial read cues, the decay mechanisms exhibited similar performance properties. With the decay mechanism enabled, the modified SDM maintained the properties of retrievals with partial cues as well as binding-error detection (not explained here). We also noticed the effects of interference. Interference was observed to be significantly higher with decay of episodes that were written (rehearsed) fewer times and related episodes which were written (rehearsed) more were retrieved when cued for the episodes that were written (rehearsed) fewer times. We now briefly review the effect of decay attributed to our two decay functions.

### Exponential Decay

The fact that episodes encoded with minimal reinforcement decayed in the $1^{st}$ and $2^{nd}$ decay cycles was ubiquitous in all tests. As expected, the rate of decay was inversely related to the level of reinforcement. Interference effects caused by reinforcement were reduced when the episodes that were not sufficiently reinforced decayed. The parameter '*a*' of the exponential decay function controlled the duration by which memory traces stayed in the system to be retrieved at some later time. The number of decay cycles by which all the memory traces decayed fully increased as the value of '*a*' increased. This is a parameter which may be domain dependent, and has to be selected by trial based on the domain and the number of dimensions to be used in the given SDM architecture.

Figures 4 and 5 illustrate the effect of the exponential decay function. In particular we are interested in retrieving events from a related episode set involving two actors, Richard and Vanessa. In figure 4, the memory is at its fourth decay cycle and is able to retrieve the last four of the patterns, with considerable difficulty in retrieving the third and fifth patterns. However, four cycles later (Figure 5, decay cycle 8), the memory system is only able to recall the last pattern. This is because this pattern was well rehearsed (repeatedly encoded) when compared to the other five patterns.

## Negated-Translated Sigmoid Decay

The results of testing the modified SDM with negated-translated sigmoid decay function were similar to what we observed with the exponential decay. The main difference was that only episodes with significant reinforcement were retrievable after several decay cycles, while all other episodes decayed at a higher rate.

```
READING : CYCLE 4

READ-LOG-EVENT
Cue:   Richard drives * Vanessa lively * Theater Friday
Status: false

READ-LOG-EVENT
Cue:   Vanessa gifted excited * wrapped present Theater *
Status: false

READ-LOG-EVENT
Cue:   * thanks sister Vanessa * wishes Theater evening
Status: true
Output: Richard enjoys  sister  Vanessa amusing onstage Theater evening
0  80.5   80.5  Richard enjoqs  sistev  Vanessa amusmne oosdage Tieater evening
1   8.5   80.0  Richard enjoys  sister  Vanessa amusing onstage Theater evening
2   0.0   80.0  Richard enjoys  sister  Vanessa amusing onstage Theater evening

READ-LOG-EVENT
Cue:   Richard enjoys sister Vanessa * onstage Theater *
Status: true
Output: Richard enjoys  sister  Vanessa amusing onstage Theater evening
0  56.0   56.0  Richard enjoys  sister  Vanessa amusing onstage Theater evening
1   0.0   56.0  Richard enjoys  sister  Vanessa amusing onstage Theater evening

READ-LOG-EVENT
Cue:   * shopped amused * modern necktie gallery evening
Status: true
Output: Vanessa selects sibling Richard unusual necktie gallery evening
0  79.0   79.0   Richard slnepe` ciqtmd  Vafhcra eoecufd necktie gallery evening
1  20.5   92.5   Vichcra slnequs sicdmf  Rafhcra uousucd necktie gallery evening
2  18.0  100.5   Vaoacsa smnects sibdin  Righar` uousuad necktie gallery evening
3  13.5  111.0   Vanessa selects sibling Richard unusual necktie gallery evening
4   0.0  111.0   Vanessa selects sibling Richard unusual necktie gallery evening

READ-LOG-EVENT
Cue:   Vanessa selects * Richard unusual * gallery evening
Status: true
Output: Vanessa selects sibling Richard unusual necktie gallery evening
0  56.0   56.0  Vanessa selects sibling Richard unusual necktie gallery evening
1   0.0   56.0  Vanessa selects sibling Richard unusual necktie gallery evening
```

Figure 4: Retrievals with exponential decay after decay cycle 4

Since the decay rate is higher for hard locations filled to about 80% of their capacity (based on the parameterization of the function), we observed that episodes written to the memory fewer times were not retrievable after the first or second decay cycle, depending on the values of the parameter '*a*'. Episodes that were sufficiently reinforced were preserved and were retrievable even after a considerable number of decay cycles.

The test-results indicate that the negated-translated sigmoid decay function filters memory traces at a higher level for consolidation to declarative memory. Episodes which were written very few to fewer times decayed away quickly due to the high initial decay rate and did not skew the retrieval of episodes which were rehearsed (written) the most. Unique episodes are rehearsed many times, hence written many times to memory. This decay mechanism shows promise in modeling transient episodic memory where only episodes which are well rehearsed are retained and hence will be available for consolidation.

```
READING : CYCLE 8

READ-LOG-EVENT
Cue:   Richard drives * Vanessa lively * Theater Friday
Status: false

READ-LOG-EVENT
Cue:   Vanessa gifted excited * wrapped present Theater *
Status: false

READ-LOG-EVENT
Cue:   * thanks sister Vanessa * wishes Theater evening
Status: false

READ-LOG-EVENT
Cue:   Richard enjoys sister Vanessa * onstage Theater *
Status: false

READ-LOG-EVENT
Cue:   * shopped amused * modern necktie gallery evening
Status: false

READ-LOG-EVENT
Cue:   Vanessa selects * Richard unusual * gallery evening
Status: true
Output: Vanessa selects sibling Richard unusual necktie gallery evening
0   56.0    56.0   Vanessa selects sibling Richard unusual necktie gallery evening
1    0.0    56.0   Vanessa selects sibling Richard unusual necktie gallery evening
```

Figure 5: Retrievals with exponential decay after decay cycle 8

## Comparison of the decay functions

We used three distinct sets of 6 associated episodes in each set, writing the memory traces with varying degrees of rehearsal. Decay was caused by the exponential and the negated-translated sigmoid decay functions. As a control, a constant decay function was also introduced. This function essentially decays the counters of each hard location by a constant (0.5 for the simulations). One memory trace in each of the 3 sets was written 120, 100 and 95 times respectively, to simulate the encoding of an episode with a significant degree of rehearsal. This was essential for capturing the properties of the negated-translated sigmoid decay function because traces encoded with limited rehearsal would decay away within a decay cycle or two.

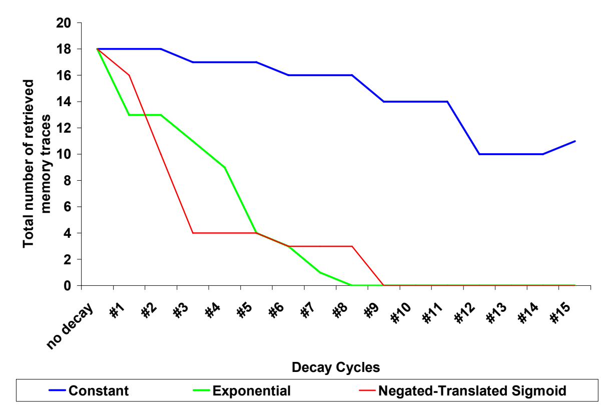


Figure 6: Comparison of the effect of decay mechanisms

The read-cues used for retrieval were partial read-cues with 1 missing feature. For each of the decay functions, the total number of retrieved episodes was computed for a system-run without decay and for 15 consecutive decay cycles. A graphical depiction of effect of the two decay mechanisms is presented in Figure 6.

The exponential decay mechanism performs quite well but has a rapid drop in the decay rate. The negated-translated sigmoid decay shows a rapid decay of the less rehearsed episodes while episodes which were well rehearsed decayed extremely slowly. These well rehearsed episodes were retrievable after several decay cycles while others were forgotten after the first couple of decay cycles. This high grade filtering ensures that only relevant, important, unique, urgent and emotionally charged episodes are retained in transient episodic memory.

## Conclusions

We have presented two possible decay mechanisms for the modified SDM system. Our simulations reveal that the negated-translated sigmoid function seems to be a more attractive model than the exponential decay function. The decay mechanisms for the modified SDM work in conjunction with interference as an effective forgetting mechanism.

## References


Albus, J.S. (1971). A theory of cerebellar functions. Mathematical Biosciences 10:25–61.

Altmann, E. M. & Gray, W. D. (2002). Forgetting to remember: The functional relationship of decay and interference. *Psychological Science, 13(1),* 27-33.

Baars, B. J., & Franklin, S. (2003). How conscious experience and working memory interact. *Trends in Cognitive Science, 7*, 166-172.

Baddeley, A. D. (2000). The episodic buffer: a new component of working memory? *Trends in Cognitive Science, 4,* 417-423.

Baeza-Yates, R. and G. Navarro. (1999). Faster approximate string matching. *Algorithmica, 23(2),* 127-158.

Brown, J. (1958). Some tests of the decay theory of immediate memory. *Quarterly Journal of Experimental Psychology*, *10,* 12-21.

Chandler, C. C. (1991). How memory for an event is influenced by related events: Interference in modified recognition tests. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 17,* 115-125.

Conway, M. A. (2001). Sensory-perceptual episodic memory and its context: autobiographical memory. In *Episodic Memory*, ed. A. Baddeley, M. Conway, and J. Aggleton. Oxford: Oxford University Press.

Donald, M. (2001). *A Mind So Rare*. New York: Norton.

D'Mello, S., Ramamurthy, U., & Franklin S. (2005). Encoding and Retrieval Efficiency of Episodic Data in a Modified Sparse Distributed Memory System. Proceedings of the *27th Annual Meeting of the Cognitive Science Society* (in press). Stresa, Italy

Ebbinghaus, H. (1885/1964). *Memory: A contribution to experimental psychology*. New York: Dover.

Fan K. & Wang Y. (1997) A Genetic Sparse Distributed Memory Approach to the Application of Handwritten Character Recognition. *Pattern Recognition, 30(12),* 2015-2022.

Franklin, S. (1997). Autonomous Agents as Embodied AI. *Cybernetics and Systems. Special issue on Epistemological Aspects of Embodied AI, 28(6),* 499-520.

Franklin, S. (2001) Automating Human Information Agents. In *Practical Applications of Intelligent Agents*, ed. Z. Chen, and L. C. Jain. Berlin: Springer- Verlag.

Franklin, S., Baars, B.J., Ramamurthy, U., & Ventura, M. (2005). The Role of Consciousness in Memory. *Brains, Minds and Media*, Vol.1, bmm150 (urn:nbn:de:0009-3-1505)

Hely, T. A., Willshaw D., & Hayes G. (1999). A New Approach to Kanerva's Sparse Distributed Memory. *IEEE Transactions on Neural Networks*.

Kanerva, P. (1988). *Sparse Distributed Memory*. Cambridge MA: The MIT Press.

Kanerva, P. (1993). Sparse Distributed Memory and related models. In M.H. Hassoun (ed.), *Associative Neural Memories: Theory and Implementation*. New York: Oxford University Press

Keppel, G. & Underwood, B. J. (1962). Proactive inhibition in short-term retention of single items. *Journal of Verbal Learning and Verbal Behavior,1,* 153-161.

Knuth D.E., J. H. Morris & V. R. Pratt (1977). Fast pattern matching in strings. *SIAM Journal on Computing, 6,* 323-350.

Lenhart, M. D., & Freeman, W. J. (1995). A Neurophysiological Explanation of the Retroactive Inhibition Memory Effect. *Computation and Neural Systems*. Monterey CA.

Lindsay, D. S., & Read, J, D. (1995). Memory, Remembering, and Misremembering. *Ptsd Research Quarterly*, 6, 1-4.

Marr D. (1969) A theory of cerebellar cortex. *Journal of Physiology, 202,* 437-470

McGeoch, J. A. (1932). Forgetting and the law of disuse. *Psychological Review*, *39,* 352-370.

Panksepp, J. (1998). *Affective Neuroscience*. Oxford: Oxford University Press.

Peterson, L. R. & Peterson, M. J. (1959). Short-term retention of individual verbal items. *Journal of Experimental Psychology, 58,* 193-198.

Ramamurthy, U., D'Mello, S. K., & Franklin, S. (2004). Modified Sparse Distributed Memory as Transient Episodic Memory for Cognitive Software Agents. In *Proceedings of the International Conference on Sytems, Man and Cybernetics*. Piscataway, NJ: IEEE.

Rao, R. & Ballard, D. (1995). Natural Basis Functions and Topographic Memory for Face Recognition, Proceedings of *International Joint Conference on Artificial Intelligence (*pp 10-17).

Reitman, J. S. (1971). Mechanisms of forgetting in short-term memory. *Cognitive Psychology, 2,* 185-195.

Tulving, E. (1968). Theoretical issues in free recall. In T.R. Dixon & D.L. Horto (eds.) *Verbal Behaviour and General Behaviour Theory*, Prentice Hall. Englewood Cliffs, N.J.

Waugh, N. C. & Norman, D. A. (1965). Primary Memory. 89-104.