Free Running Subsumptive Regular System (SRS22)

Richard Keene Jan 1, 2022

A Free Running Subsumptive Regular System (SRS) is described, and several conjectures are presented. The goal is to establish an approach to Artificial General Intelligence (AGI) that is self-directed, has aspects of cognition, and can be implemented on current systems. Cognitive behavior is extrapolated. The system is theorized to exhibit behaviors of self-clocking, neural avalanche, inner vs. outer attention, reacting to the environment based on experience and more. This paper is a refinement and reworking of Artificial Cognition, from the conference Intelligence in Neural and Biological Systems, 1995. INBS'95, Proceedings (Keene, 1995) (<https://dl.acm.org/doi/10.5555/850955.854064>).

Table of Contents

[2 Definition of Terms and Information Flow 3](#_Toc103679167)

[2.1 Terms 3](#_Toc103679168)

[2.1.1 Subsumption 4](#_Toc103679169)

[2.2 Loops in the System 5](#_Toc103679170)

[2.2.1 Lockup and Unlock 5](#_Toc103679171)

[2.3 Free Running Emergent Behaviors 6](#_Toc103679172)

[2.4 The Importance of Random Noise 6](#_Toc103679173)

[3 Match Selection 7](#_Toc103679174)

[3.1 Selecting Best Match to M 7](#_Toc103679175)

[3.2 New Patterns 7](#_Toc103679176)

[3.3 When to Learn 7](#_Toc103679177)

[4 Concept Maps 8](#_Toc103679178)

[4.1 Attention: Internal and External 8](#_Toc103679179)

[5 Multi-SRS systems 9](#_Toc103679180)

[5.1 System Creation 9](#_Toc103679181)

[6 Global Pattern Connectivity 10](#_Toc103679182)

[6.1 Connectivity Triples ⅄ 10](#_Toc103679183)

[6.2 Meta-Patterns and Anonymous M 11](#_Toc103679184)

[7 Revised Diagram 12](#_Toc103679185)

[8 Implementation Notes and Conjectures 13](#_Toc103679186)

[8.1 Synchronization 13](#_Toc103679187)

[8.2 Neuron Pulse vs. Value 13](#_Toc103679188)

[8.3 Redundancy 13](#_Toc103679189)

[8.3.1 Pattern Matchers 13](#_Toc103679190)

[8.3.2 Multiple Muti-SRS systems 14](#_Toc103679191)

[8.3.3 Biological implementation of G 14](#_Toc103679192)

[8.3.4 Derivation of X, Y, Z coordinates 14](#_Toc103679193)

[8.3.5 Connectivity Triples 14](#_Toc103679194)

[8.3.6 Distance from the Spinal Cord 14](#_Toc103679195)

[9 References 15](#_Toc103679196)

# Definition of Terms and Information Flow

World and Hardware  
and/or   
Other SRS

F (Transform Function)

I (Inputs)

Change Rate

O (Outputs)

World and Hardware  
and/or  
Other SRS

G (Goodness Function)

Previous Patterns

Previous Patterns

Previous Patterns

[P] Previous Patterns

S (State)

S’  
(New State)

Q   
(Match Quality)

C Match Selector

Regular Pattern Matching System

M  
(Concept Map)

OF (Output Transform)

Subsumptive System

System State

Figure 1 - SRS System Diagram (One SRS Module, also called SRSUnit)

## Terms

The sensory information **I** is fed into a preprogrammed function **F**. The output of **F**,and the average rate of change for the sensory information , are fed into **M**, the Concept Map. The concept map is the input to an (optional) output transform function **OF** and operates some hardware out to the world. This is the hardwired subsumptive system. By hardwired we mean to say there is no learning or change over time.

The Regular Pattern Matcher is “regular” (as in “orderly”) in that it is an array of learned patterns and simple matching and learning. The matcher and patterns exhibit learning and change over time. The pattern matcher looks at the current state **S** of the subsumptive system and compares it to all learned patterns. The best match **S’** and the quality of match or strength of match **Q** is used to stimulate or inhibit **M** such that

Eq1:

(Element by element multiply, not matrix multiply).

The Match Selector **C** has a associated with matching and learning new patterns. While patter matches are immediate, the delay between seeing a pattern and using **G** to decide to learn the pattern is significant in system operation.

The goal of computing **S’** : At time **T**, predict the future state of **M** at (and thus indirectly the future state of the external environment) and react earlier (by and in advance of the actual environmental state to gain an evolutionary and survival advantage. It is hypothesized that reacting to the future environmental state is also the goal of all cognitive systems that have any ability to learn.

The Goodness Function **G** evaluates the environment for how well a goal of survival or other goal is currently being met. **G** is derived from, and establishes, the reason the AGI exists. It is likely that a more nuanced Goodness function will be developed.

**[P]** is a series of previously seen patterns. A single pattern in **[P]** is called **P**. A single element in the pattern **P** is **p**. For simplicity this is discussed as if a single entry in **[P]** is a direct copy of a previously seen **M**. See section 6 for a more nuanced definition of **[P]**.

The Match Selector **C** (for Comparator) constantly compares **S** with a collection of previously seen patterns **[P]** and selects the closest match **S’**. **C** also looks at **S** and compares to all **[P]**. If there is no close match, and goodness is increasing, **C** records M as a new pattern. **C** only records patterns that are sufficiently different from any of **[P]**. The actual process is complex and described is section 3.

An **SRS Module** is one instance of Fig. 1. A full system will have many modules and a large variety of **F** and **M**. The interconnection of modules forms the Subsumptive system. The core design of the full system focuses on **F** and interconnectivity of Concept Maps.

The term Concept is used here because the map **M** represents some abstraction or concept of the environment or further abstraction of up-stream Concept Maps. As a concrete example a low-level concept map (or just “map”) might take raw video frames as input, and extract edges. The next map (next downstream SRS) might take the edges and extract edge orientation around the center of the image frame.

### Subsumption

One concept mentioned herein is **subsumption**. Subsumption (Thompson, 2009) (Brooks, 1985) (Wikipedia, Subsumption Architecture, 2021) is to place something in a larger context such that the lesser context is encompassed. In our use herein we mean to take simple functions that are stimulus-reflex and provide simple behavior. We then override or subsume them when some situation arises where more complex and nuanced behavior is needed. The lower-level behavior is temporarily dormant while the higher-level behavior is manifest.

#### Subsumption and the Need for Forward Additive Neurons

In a directed analog network (consisting of Op-Amps and resistors) one can take inputs and combine them and generate outputs. The classic example of this is “Photovores” where a small robot seeks light sources. One can use subsumption and generate rather complex analog networks to arrive at complex behaviors with very biological looking movement. The problem with evolving such a system is that one change anywhere in the analog network upsets the entire network and the network’s resistors need to be recalculated. If a random new path is added (mutation) without recalculating the entire network, the chance of having a functional system is very near zero.

With neurons that are pulsed, if the new mutation’s neurons are not active the neurons in effect do not exist. One can refer to this as **neural hiding**. (Not related to “hidden layers” in neural networks.) A random mutation to the pulsed neural network has a larger probability of not being completely destructive to the system operation and survival and might even be a survival benefit. In a computer system we add values forward into the next SRS representing pulse rates. See section 7.2 for more on pulse vs. value.

Using subsumptive design and implementation strategies results in a system that can be designed and modified much more easily. One can incrementally refine the overall system. One can even snapshot the entire state of the current system and learning, and then modify the system and add or remove SRS and reload the system and continue. This is a trick that evolution cannot do and instead must produce a new generation to test mutations.

## Loops in the System

There are several feedback loops in the system.

O (Outputs)

World and Hardware  
or  
Other SRS

F (Transform Function)

I (Inputs)

Change Rate

G (Goodness Function)

Previous Patterns

Previous Patterns

Previous Patterns

[P] Previous Patterns

S (State)

S’  
(New State)

Q   
(Match Quality)

C Match Selector

Regular Pattern Matching System

M  
(Concept Map)

OF (Output Transform)

Subsumptive System

World Interaction Loop

World and Hardware  
or   
Other SRS

System State

Figure 2 - Feedback Loops

The first and most obvious loop is the World and System interaction shown in light gray. This type of interaction had been discussed in detail by other authors. See (Chan, 2001) for just one example.

The much more interesting feedback in the system is between the Match Selector(s) **C** and the Concept Map(s) **M**. With high and low **Q** the system simply acts as a subsumptive network with no learning. This is simple stimulus, transformation, and reaction.

In the case of a static (boring) environment is low so in effect **F** is turned off. The cycle of **C** matching, selecting one of **[P]** and setting **M** to that pattern (possibly subdued by **Q**) and then repeating the cycle will cause a **free running** cycle through related patterns. The loop also causes a virtuous cycle of match, weakly stimulate M, and repeat. This will cause a vague pattern match to **avalanche** into a definite pattern. Somewhat related reference is (Yi Li, 2018). Also see

### Lockup and Unlock

The **CM** loop has the potential to “lockup” in a single state or a repeating chain of states. There are several mechanisms to prevent this. First is that selected patterns are marked “fatigued” and have greatly reduced selection probability, that decays over time. The decay interval is directly related to the of **C**.

Another mode of breaking out of pattern lockup is when the environment changes and **F** gets precedence over **C**. Interestingly, this may not happen in a single tick of time, but can also be a gradual avalanche, and both **C** and **F** can be blended into an output response and in effect switch to some other chain of patterns; “switch to another train of thought”.

A third case is that the SRS can be in a subsumptive and hierarchical system of other SRS and this SRS can be the input to other SRS and can also have other SRS as its input. Each SRS has its own and all SRS run asynchronously so another SRS could break this SRS out of a lockup. This would happen when the upstream SRS has an abrupt change and rises. See section 5.

## Free Running Emergent Behaviors

The system free runs in the above loops and each SRS is asynchronous and has its own . The SRS operates on a **tick** that is some fraction of .

Some SRS may be considered as “low level” modules with short such as in the above edge detection example.

Other much longer SRS that are “higher level concepts” with longer . High level SRS may take longer to react to external stimulus and thus the situation can emerge where the low-level SRS are “in the real world” and high-level SRS in a state more related to **[P]** matching and avalanches. Thus the high levels are on an independent or semi-independent train of thought not directly related to the current world state.

There will be a fuzzy boundary relative to time scales that will change moment by moment. Thus, the system may maintain external robotic control while at the same time be strategizing what higher level actions to predict toward and act on.

## The Importance of Random Noise

The system will run in **CM** cycles and respond to the environment as programmed. It is useful to have occasional completely random match hits in the matcher if both and **Q** are low. High-level concept maps should have this behavior, though it should be very infrequent. This disturbs the system out of idleness. This also give the system a propensity to think, rather than be idle. One can have this randomness occur more often in a calm and idle system. Very calm environments will allow the system to self-process and possibly make subtle connections of concepts and relationships that otherwise would be missed. This also would be processing that would cement the identities of **Ṁ**. See section 6.2.

# Match Selection

The Match Selection algorithm does several operations.

## Selecting Best Match to M

The array of past patterns gets scanned for which is the closest match to **M**. The closeness of match also generates a **Q** quality. Each member of **[P]** has a last use time and selection probability goes up as the time increases. This is referred to as **neural fatigue** and probably equates to the refractory period of neurons with a short or, for longer SRS, to other biochemical processes. This is difficult to pin down as a concept in the literature due to refractory periods being considered more of a limitation than as a constructive effect. In the SRS design the is critical to sequential free running pattern matching and chains of patterns.

## New Patterns

If Match Selector gets a poor match on **M** then it will copy **M** into **[P]**. Obviously, this will cause infinite growth of **[P]** so we limit how large **[P]** can grow. If **[P]** is full then the Matcher will find the two most similar patterns in **[P]**, average them together into one entry and use the other entry for the new **M**. A possible extension of this algorithm could also keep how stale an entry is in **[P]** (how long since it participated in a successful match) and factor in staleness in deciding which pattern to eliminate if very stale or allow it to be considered more easily as most similar to another pattern. Thus, the number of patterns the system can learn is directly related to the size of **[P]**.

The check for the two most similar patterns in **[P]** can also be accelerated if the search is semi random and we only want a fairly near match. This slight randomness in selection is desirable as there may be no one right answer or prediction of future state.

## When to Learn

Eq 2: Learn Probability: **L = {>0│0}**

The matcher should not try to learn every single pattern that is ever in **M**. Rather it wants to learn patterns the in the future cause higher **G**. Thus, the derivative of **G** averaged over , if increasing indicates a good pattern to learn. Since an event at **T** +  (where **T** is “now”) may have a very poor **G** or decreasing **G** at **T**, we want to look at **G** nearer to **T +** , one possible time frame would be the average

Eq 2:

of **G** as in Eq 2. Since we need time dependent **G** values a fifo (first in first out buffer in this case at ¼ intervals) must be kept for this algorithm. An alternative to a fifo might be a long and short term **G** buffer (two single values) that follow **G** averaged over two time spans. This also would give the derivative easily.

**L** is not a learning rate as in traditional back propagation. It is the probability of learning by recording a snapshot of **M**.

# Concept Maps

A SRS is also called a Concept Map. The map **M** is the current representation of some abstraction of data. This could be something simple and immediate like stereo sounds direction, or something complex and longer term like overall movement or travel goal. The map **M** could also be any dimension as an array including 0 (a single value), 1 a linear array e.g. stereo sounds direction, 2D or higher dimensions.

## Attention: Internal and External

The map blending equation (Eq 1)

Eq 1:

Is implemented such that on each tick of the SRS **M** gets updated. We use a SRS tick interval so as to not have to update **M** for every SRSon every cycle or tick of the overall SRS System. This lets one distribute the processing of longer SRS.

The attention can be external with high or internal with high **Q**. If is high then the SRS is acting purely as a subsumptive directed graph of functional transformations. One can force high when designing and debugging an SRS. With high **Q** the SRS is imagining some environmental state. Forcing **Q** high and low generates an autistic system with no external attention. It is interesting when **Q** and are balanced. They system is then generating actions based on past experience blended with the current environment and thus has volition. The blended **M** represents a solution that will tend toward higher **G** in the future given the current environmental state. This balance between **G** andis key to a useful operational system.

# Multi-SRS systems

## System Creation

The overall system is designed by intuiting what mappings of the environment might be useful to attaining a higher **G**. A useful **F** would be any transformation that arrives at some abstraction of what is going on in the environment that might promote survival and a higher **G**. One can name particular **F** as a terminology for system design.

As an example, one might generate an SRS called “Sound Approaching” that takes the derivative of audio input volume in stereo, and direction, and maps to a 1D array of 360 values representing how much sound is increasing from a given heading in degrees. Other SRS then can use Sound Approaching to compute further abstractions or generate reactions such as fleeing from the approaching sound.

With a subsumptive design one can then design another SRS that is “Pheromone Sensing” (maybe our robots emit a weak radio signal to indicate reproductive readiness) and overrides Sound Approaching. This makes the robot flee sounds unless there is a signal of reproductive readiness in which case it seeks the sound instead. Thus, one can design freely and connect the SRS network to achieve hierarchical behaviors without having to invent the entire system at once.

One could also design the system with two independent multi-level SRS systems with different goals, and some limited inter communication (each system is tied into a relatively few SRS in the other system) and have an overall system that “thinks” in two modes and thus a “split brain”. The common SRS between the two systems would be low-level and the separate system SRSs would be more high-level. Not being limited by evolutionary history one could design a system with many such sub-SRS-Systems.

# Global Pattern Connectivity

The SRS design described above uses **[P]** as an array of copies of previously seen **M** for conceptual simplicity. It is assumed that a given **M** at timer **T** can be used to predict the future **M** at time **T +** . This may not actually be true. For example, a current **M** representing the center of image edge angle will not very well predict what the edge orientation will be at **T +** . Such a prediction requires a more global pattern match.

A better predictor is to snapshot the entire multiple SRS systems for a global pattern match. Obviously, this would be such vast amounts of data that one could not store it and a biological brain can not simply have a few million extra copies if itself. A much more realistic approach is to have an entry in **[P]** not be just a copy of **M** but also some percentage of links to other **M** in other SRS. This extrapolates to a new definition of **[P]**:

**[P]** is an array of sets **P** of SRS Pointer:Value pairs **p** representing a sparse snapshot of the entire system with bias toward the current SRS concept and related concepts.

* To expand on the above definition, “with bias toward the current SRS concept” means related semantic concepts. This then indicates a definition of conceptual distance which is encoded as a X, Y, Z physical distance for convenience. In a biological brain, sections contain related concepts. For example, audio processing (hearing) is in a localized area of the brain. Motor functions are in a different area. When designing a multi-SRS system each SRS can have a X, Y, Z, conceptual coordinate. When semi-randomly selecting a single reference in a single **P** in **[P]** one biases toward lower Euclidian or Manhattan distance. This sparse snapshot of the entire system biased toward the concept of interest is a proxy for “the entire state of the multi SRS system”. Expecting this prediction of future state to be correct is quite reasonable. See note 7.3.4. See (Wikipedia, Talairach coordinates, 2020) and references therein.

## Connectivity Triples ⅄

If a single SRS has a pattern that is 10% of **M** and then another 75% of fairly closely related concepts, and the remaining 15% of connections being to pointers in the entire global system, one could arrive at useful predictions. This series 10:75:15 (always adding up to 100) is a **connectivity triple** and is represented by the symbol **⅄** an upside down Y representing 3 parts. The three elements are **⅄1,** **⅄2, ⅄3.** One could posit connectivity quadruples or more elements. The triple seems to communicate the intended connectivity well.

Interestingly, the **⅄** of is a representation of local vs global connectivity and directly relates to the type of predictions made. A **⅄**of 50:49:1 will give predictions that are immediate and concrete. A **⅄** of 5:5:90 will be a very global holistic prediction. Different concepts (SRS) should have varying connectivity triples.

This new definition of **[P]** also changes the definition in section 3.2. We can make a choice in algorithm whether to track stale state of an entire entry in **[P]** or track staleness on a single pattern pair element basis. Thus, **p** can be tracked based on their usefulness to correct pattern matching. This is reminiscent of how the brain uses sleep time to destroy unused connections. This high granularity of stale-tracking may not be necessary for a successful SRS system.

In biological systems local connectivity requires shorter neurons than far connectivity. So **⅄** with large **⅄3** are expensive but also are very valuable for overall pattern matching and prediction, and thus high IQ. Due to evolutionary history, the overall pattern matching and in particular **⅄** with large **⅄2** were probably added rather late. Thus the **[P]** implementation in a biological brain has **[P]** and the pattern matcher as an added layer on the outside of the multi-SRS survival-reflex system and is the cortex. Each cortical neuron then has a target area deeper in the brain that it is trying to predict and stimulate the inner brain sections into the future state. One then would expect cortical neurons to have vertical long connection into the inner brain as observed.

## Meta-Patterns and Anonymous M

One interesting effect falls out of this description. If the matcher is trying to predict the future **M**, one could also have matchers that are trying to predict the future matches of other matchers. In effect the matcher becomes a meta-SRS and very abstract concepts that have no direct environmental correlates could get set up in the Matchers. This refactors into the concept of an SRS that has no **F** but is simply an **M** and is an anonymous concept **Ṁ** (M with a dot over it). This is not a designed SRS and **F**. One simply has a large array of initially identical SRS **M**. Over time and learning these **Ṁ** begin to represent abstract concepts. This might correlate to the cortex processing abstract ideas like “loyalty” or “chi”.

# Revised Diagram

World and Hardware  
and/or   
Other SRS

F (Transform Function)

I (Inputs)

Change Rate

O (Outputs)

World and Hardware  
and/or  
Other SRS

G (Goodness Function) FIFO

Previous Patterns

Previous Patterns

Previous Patterns

[P] Previous M

S (State)

S’  
(New State)

Q   
(Match Quality)

C Match Selector

Regular Pattern Matching System

M  
(Concept Map)

OF (Output Transform)

Subsumptive System

Previous Patterns

Previous Patterns

Previous Patterns

[P] Previous Patterns, **⅄**

Other SRS

Reaper

Global  
Goodness

 sss

Figure 3SRS Module also called SRSUnit - Revised

# Implementation Notes and Conjectures

It is imagined that implementing a single SRS and an entire SRS system would be done using CUDA on one or more GPUs.

## Synchronization

The various SRS do not need to do any locking or synchronization. In the best case a GPU thread that wants to modify a **M** can just do so and since the total elements of all **M** is very high compared to the number of threads, there will be few collisions. In the case of a collision, the reader of the **M** might get a previous value or even a corrupt value. On the next SRS tick, it will get the right value and the error is just slight noise in the overall system.

Note: The actual implementation currently is two phase per system tick. A “compute next state” and then “copy next to current state”.

One would synchronize the GPU on every simulation tick (not SRS tick) to copy in and out external world inputs and outputs.

Also, the reaping of stale connections and **M** would have to be on a sweep basis. This is quite like sleeping and cleaning up neural connections. One disables direct muscular activity so one does not thrash about in sleep. So, one would expect the system to occasionally halt for cleanup and garbage collection.

## Neuron Pulse vs. Value

In biological systems one evolutionary reason for neurons being pulsed is that the pulse rate can be relatively immune to oxygen, sugar, or other factors. Pulsed neurons also enable some time critical short-term computation and measurement such as sound direction. Pulse summing enables neural hiding, which is a important feature of the system.

In a computer simulation short time critical events can be accounted for by having **F** that directly compute the time differences. Subsumption can also be achieved with simple addition of values rather than actual pulses. **F** and **OF** are optional. The implementation has all maps as subclasses of SRSUnit being the generic Concept Map module. Some SRSUnit are of the same class if they have general input connections to other SRSUnit and not to hardware IO.

## Redundancy

The redundancy to compensate for cell death and damage is not needed in computer digital system. Thus, it is expected that an SRS could achieve AGI equivalent to the human brain with an order of magnitude fewer elements and connections. (1% as many neurons and connections?) See (Glassman, 1987) for an interesting read on redundancy. Also (FABRIZIO DE VICO FALLANI, 2012) Biological Correlates

Here are some random notes on biological systems that do not fit cleanly in other sections, or that are even less than conjectures.

### Pattern Matchers

In an SRS the pattern matcher and **M** are seen as objects. **M** is an array of values or neurons, but the pattern matcher is a programming construct and does not appear anywhere directly in the brain. Rather, the pattern matching algorithm is built directly into each neuron and is an emergent behavior.

### Multiple Muti-SRS systems

The brain is obviously split into two hemispheres, the left and the right. The hemispheres operate as fairly independent thinkers but with two styles. No, the author will not start making grandiose statements as to what each half does. See (McGilchrist, 2012).

### Biological implementation of G

Is **G** the equivalent todopamine in organic brains? It has been shown that dopamine increases cause learning. An event followed by a reward is the fastest teaching technique. The technique of event, then punishment, then reward after changed behavior has been shown to be less effective in psychological studies.

### Derivation of X, Y, Z coordinates

One could use the X,Y,Z location of sections of the human brain as a guide for SRS concept coordinates. In a program SRS can have the same X, Y, Z where in a biological brain, neurons must be spatially separate. Thus, one could use the location of hearing in the brain coordinates in millimeters or Talairach Coordinates. This also might get the closer to human though reproduction and a system one can understand. (Wikipedia, Talairach coordinates, 2020)

### Connectivity Triples

Connectivity Triples are a simplified method to talk about how connected some section is, and to encode that connectivity into program meta-code. In a biological brain the effect of triples would be embodied in how aggressively cortical neurons connect further away vs. nearby and would be a continuous curve of probability to connect at a given distance.

### Distance from the Spinal Cord

One might conjecture that neural paths close to the spinal cord have short and the furtheraway sections represent more abstract concepts and longer .

## Avalanche and Generative Adversarial Networks and/or Autoencoders

There is some similarity between avalanche into some system state and GANs (Goodfellow, 2014). In particular, if the output of a GAN is fed into the input of itself the GAN will then constantly generate new versions of its scene or domain. In effect the GAN generates a scene then cascades onto some other scene ad. Infinitum.

An effect that looks very similar (but is not an avalanche) is interpolating in latent space directly. (See https://www.reddit.com/r/deeplearning/comments/urbfv8/latent\_space\_interpolation\_of\_an\_ascii\_art/)

# References

Brooks, R. (1985). A Robust Layered Control System for a Mobile Robot. *Massachusetts instuitute of Technology*, 25.

Chan, S. (2001, October 31). *Complex Adaptive Systems.* Retrieved from web.mit.edu: https://web.mit.edu/esd.83/www/notebook/Complex%20Adaptive%20Systems.pdf

FABRIZIO DE VICO FALLANI, J. T. (2012). REDUNDANCY IN FUNCTIONAL BRAIN CONNECTIVITY FROM EEG RECORDINGS. *International Journal of Bifurcation and Chaos, Vol 22, No 07*.

Glassman, R. B. (1987). An hypothesis about redundancy and reliability in the brains of higher species: analogies with genes, internal organs, and engineering systems. *Neurosci Biobehav Rev*.

Goodfellow. (2014). *Generative Adversarial Nets.* Proceedings of the International Conference on Neural Information Processing Systems (NIPS 2014).

Keene, R. (1995). *A New Model for the Cognitive Process.* IEEE, INBS '95: Proceedings of the First International Symposium on Intelligence in Neural and Biological Systems (INBS'95).

McGilchrist, I. (2012). *The Master and his Emissary – The divided brain and the making of the western world.*

Michael Pereira, D. P. (2022). A leaky evidence accumulation process for perceptual experience. *Trends in Cognitive Sciences*, 11.

Thompson, T. (2009). Improving control through subsumption in the EvoTanks domain. *Proceedings of the 2009 IEEE Symposium on Computational Intelligence and Games, CIG*.

Wikipedia. (2020). *Talairach coordinates*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Talairach\_coordinates

Wikipedia. (2021, Nov). *Subsumption Architecture.* Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Subsumption\_architecture

Yi Li, G. W. (2018). DeepIM: Deep Iterative Matching for 6D Pose Estimation. *arXiv:1804.00175*.