Learning timing in real-time using adaptive drift-diffusion processes

François Rivest

March 22, 2011

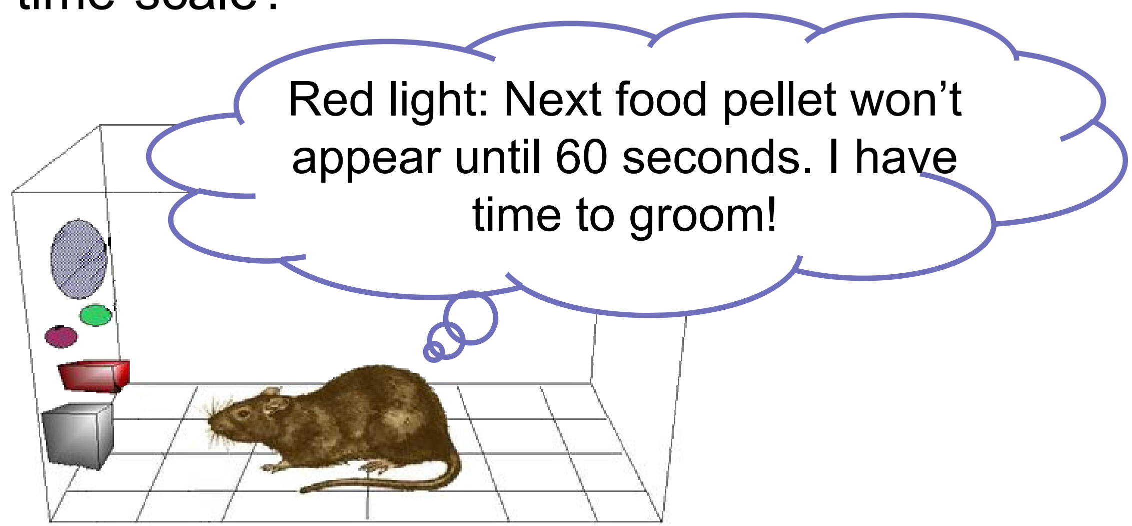
Barbados

Plan

- The problem of multi-seconds time representations
- Standard approach and some limitations
- Animals' data suggest
  - Let's change the learning axis
  - Let's learn timing faster
  - Adaptive drift-diffusion processes
- Preliminary results & future work
- Conclusion

The problem

- Predicting the future:
  - How do animals rapidly develop representations that predict up coming events from tenths of seconds to multi-seconds time-scale?



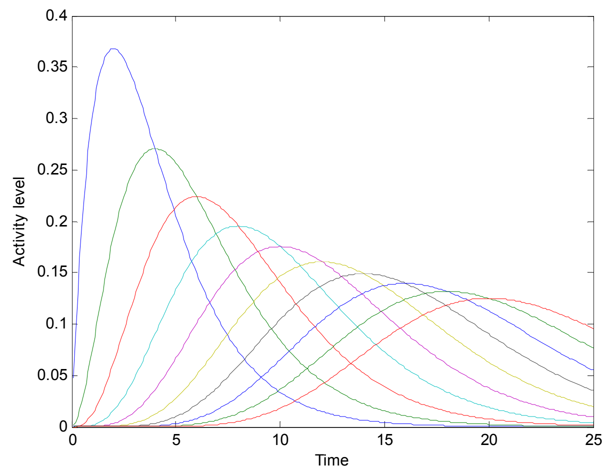
Standard approach to the problem

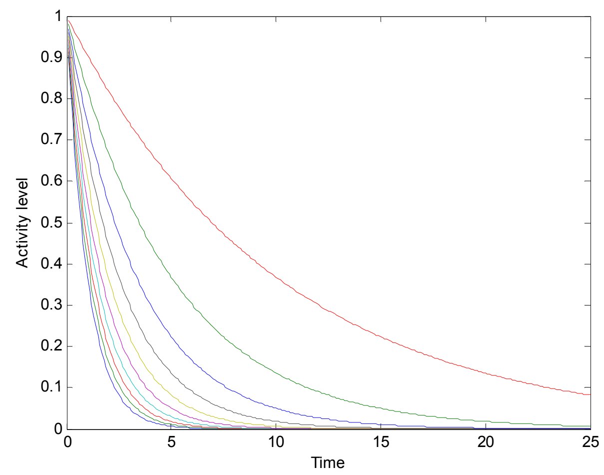
- Add an appropriate prebuilt population of temporal functions with predetermined timing distributions.
  - The population must be generated such as to fit the necessary temporal resolution and range.
  - There must be a distinct population for each stimulus that needs to be temporally represented.
  - Often, start signal (usually stimulus onset) must be known a priori.

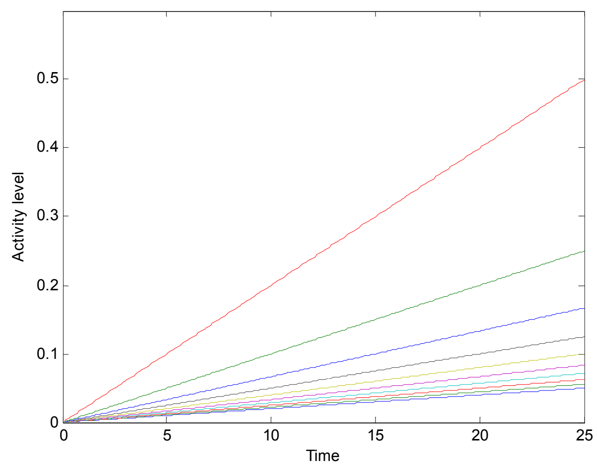
- The memory and computational requirements of such representations can rapidly explode!

Usual temporal representations

- Diffused delay-lines
- Exponential decay processes
- Linear build-ups (ramps)







Learning temporal representations

- Recurrent neural networks can represent elapsed time in a manner similar to real neural populations (Rivest&al.2009).
- But recurrent neural networks are relatively bad at learning relationships between events distant in time (Bengio & al. 1994, Hochreiter & Schmidhuber 1997)

- <u>Compared to animals</u>, the number of trials to learn a time interval gets huge as the distance (in number of time-step) increases between the events. (Rivest & al 2009).
  - Thousands of trials for LSTM compared to few dozens for animals.
- And temporal noise (noise in the timing) makes it worst.

HOW COULD ANIMAL DO IT?

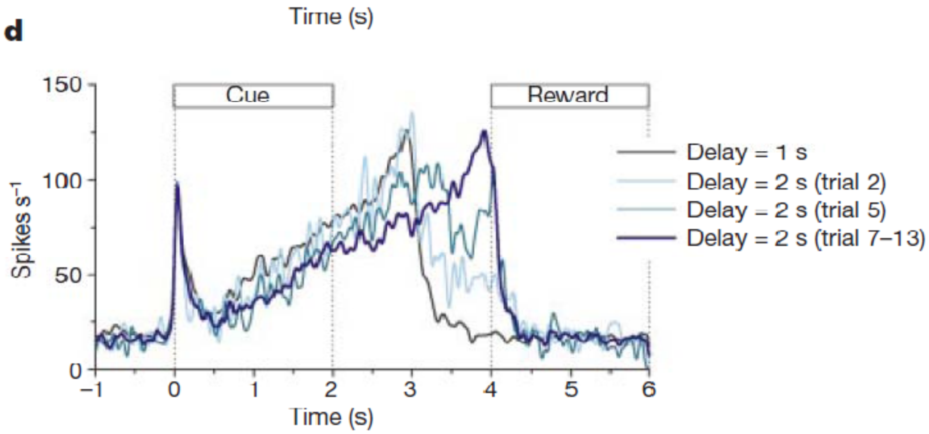
The modeling approach

1. Look at animal data
   - Behavioural, electrophysiological, etc..

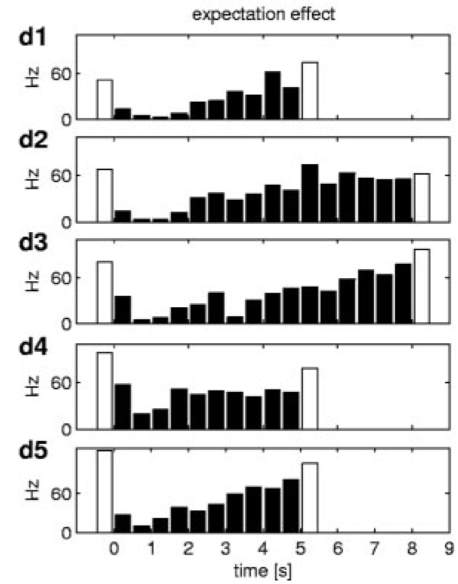
2. Examine what kind of computational properties the animals ‘algorithm’ seems to have.

3. Use them to explore their possible learning algorithms:
   - Eg.: for real-time learning of interval timing.

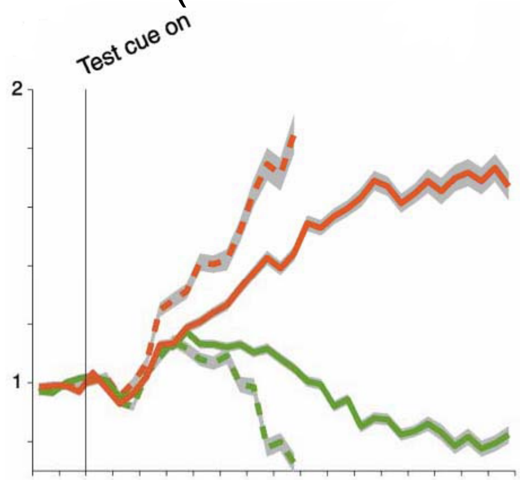
Observed neural data



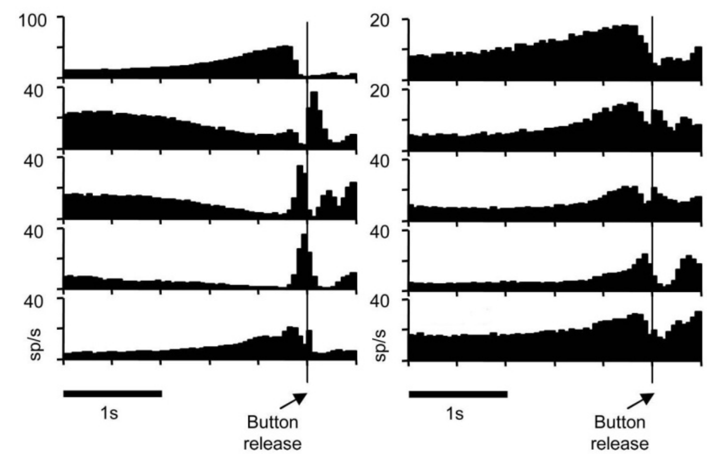
Thalamus (Komura & al. 2001)



IT (Reutimann & al. 2004)



LIP (Leon & Shadlen 2003)

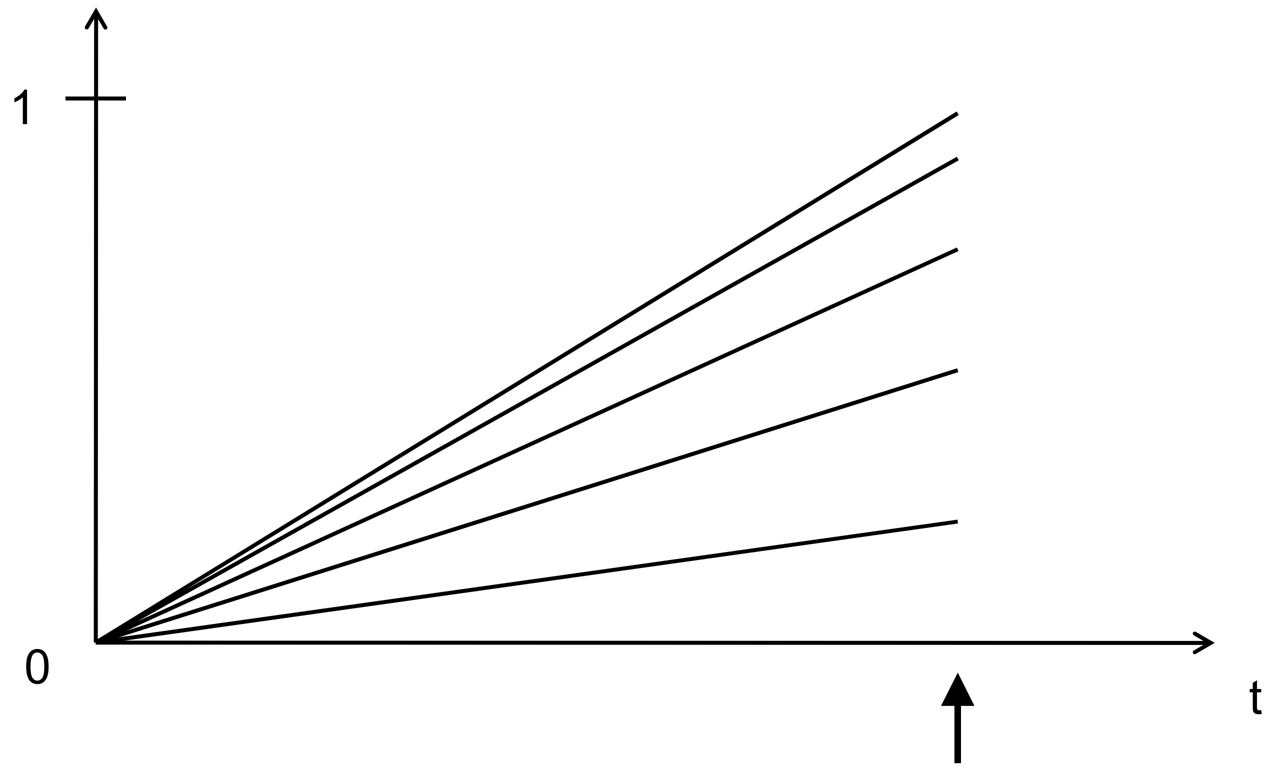


M1 & PMd (Lebedev & al. 2008)

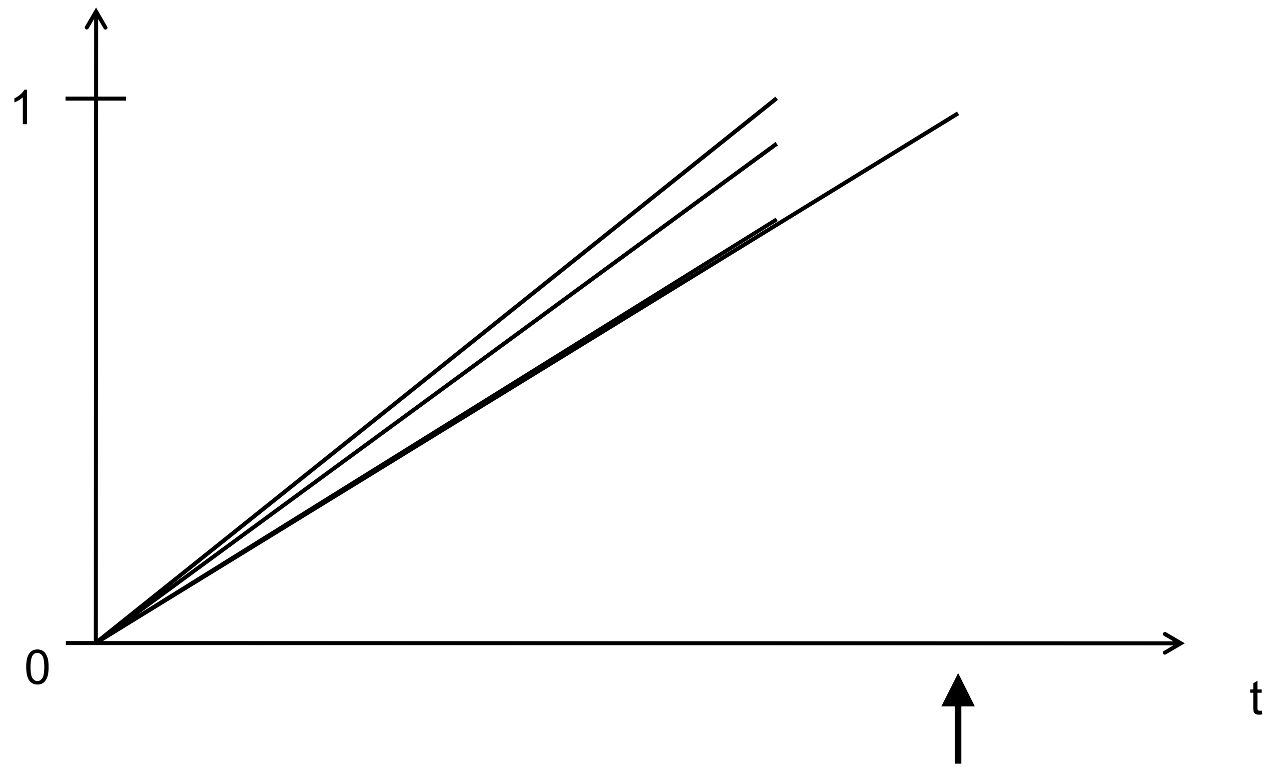
Observations

- Neural activity is naturally bounded
- Elapsed time seems to be accumulated
- Its slope seems proportional to the delay encoded
- Its slope adapts to time interval when it changes
  - As opposed to fixed, prebuilt population, temporal code.

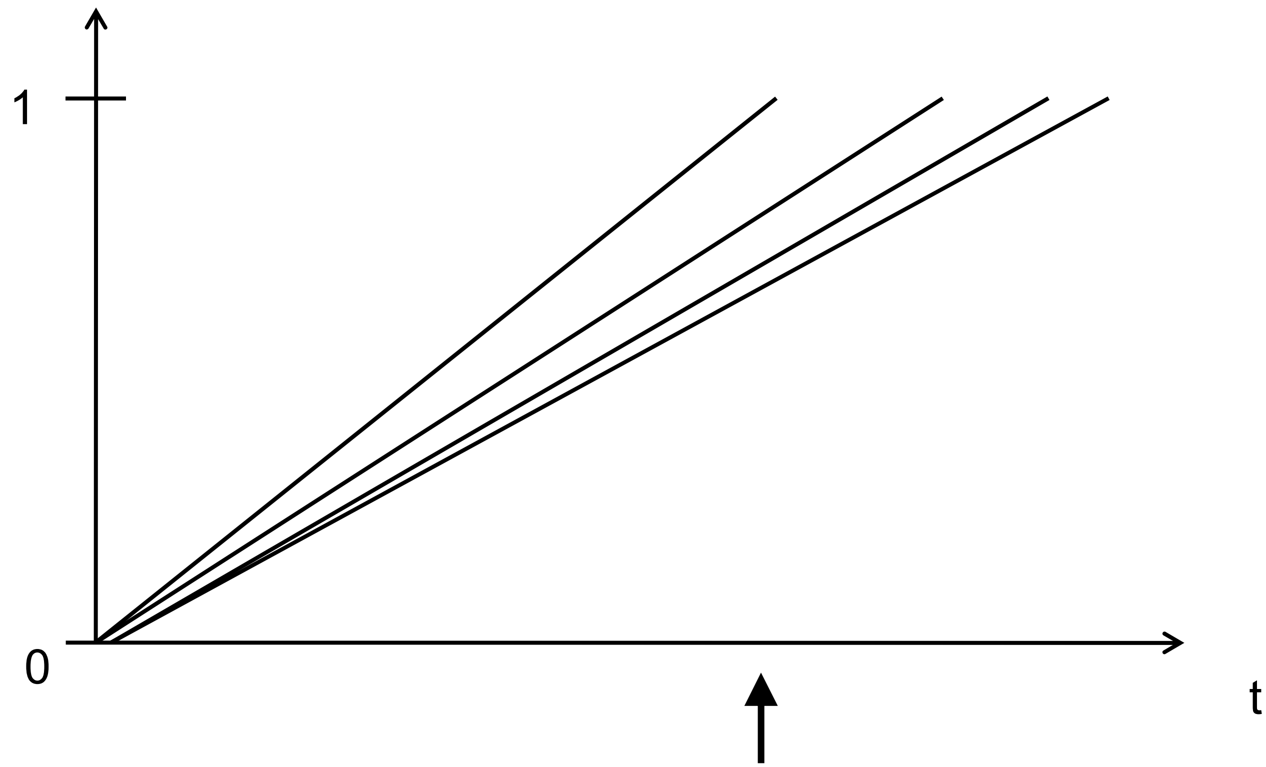
What we are looking for



What we are looking for

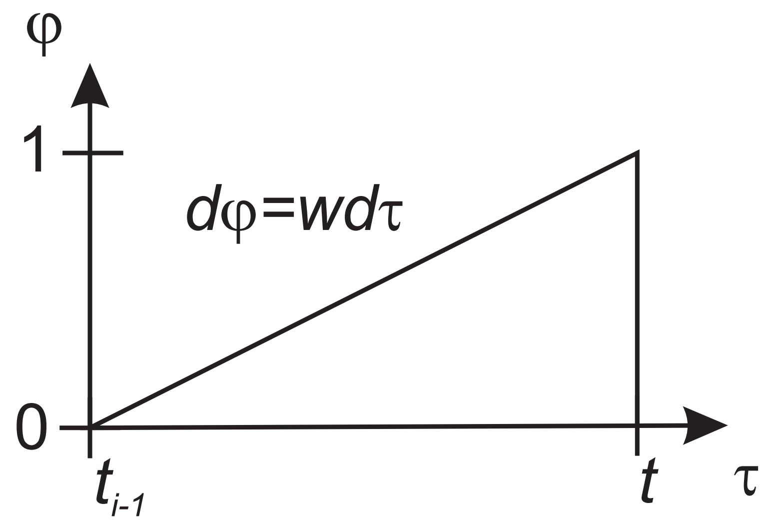


What we are looking for



Temporal integration

- Single stimulus *x*
- Regular event *y*
- $d\varphi = 1_{\varphi \neq 1} xwd\tau - y\varphi d\tau$
- *w* = rate of event *y* given *x*

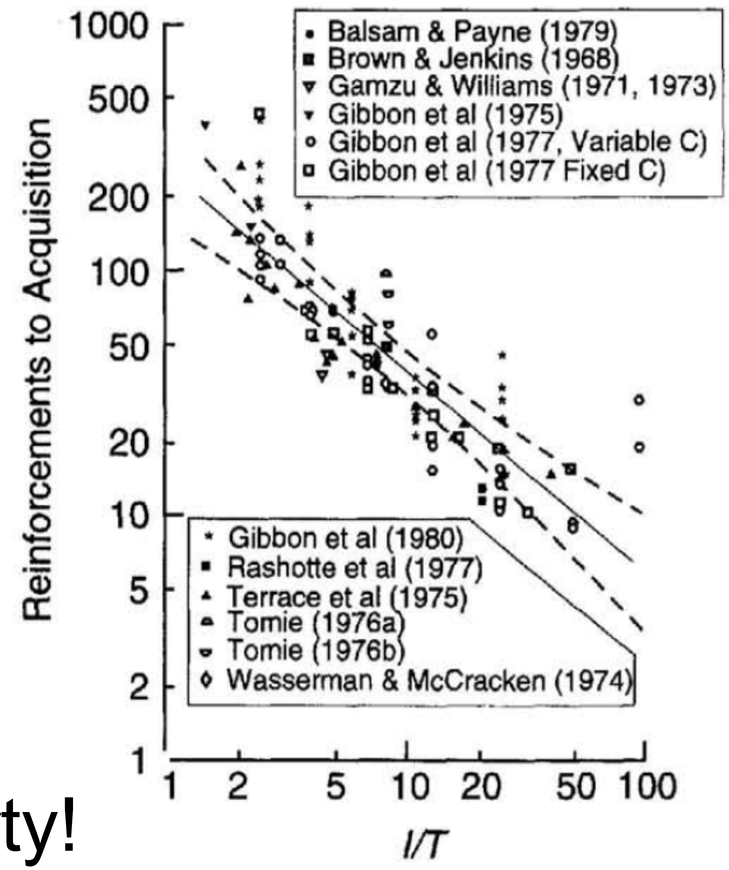


$$\varphi(t) = \int_{t_i}^{t} 1_{\varphi \neq 1} xwd\tau - y\varphi d\tau = \min\{(t - t_i)w, 1\}$$

- If *y* is a reward, *w* is the reward rate for *x*
- Elapsed time is given by *φ/w*
- Expected time is given by *(1-φ)/w*

How fast should we learn?

- Animals learn the timing as fast as the association
- The learning speed seems time-scale independent
  - Even if the time-scale is 10x bigger, the number of trials to acquire a timed response is similar!

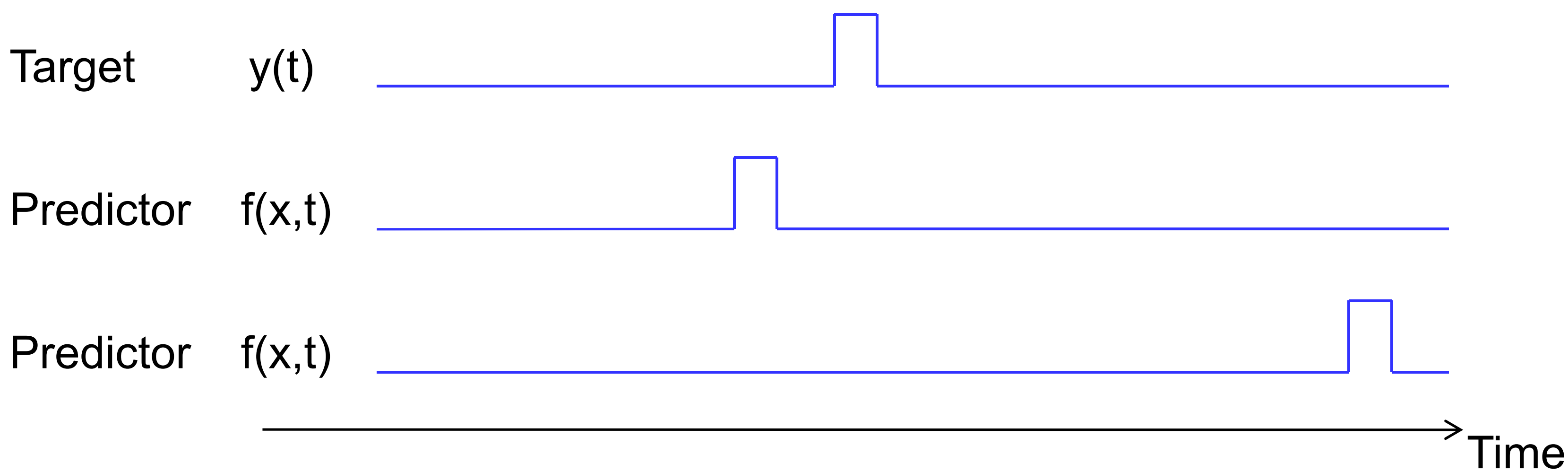


- Definitely not a RNN property!

(Gallistel & Gibbon, 2000)

Changing the axis of error

- Are these two prediction errors of the same size?

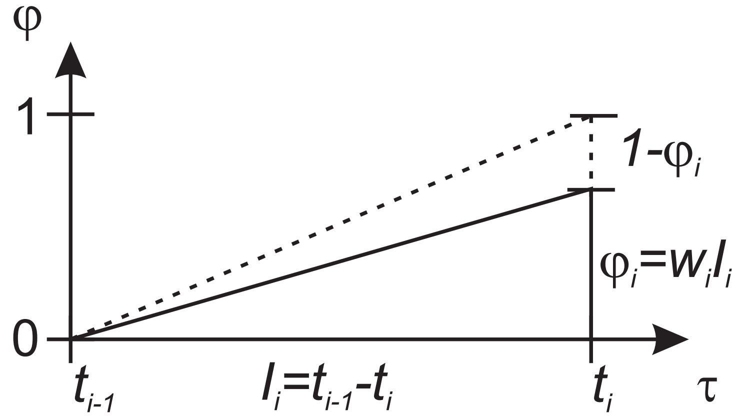


- Let's minimize $(t_{f(x)} - t_y)$ instead of $(f(x_t) - y_t)$!

Integrator time learning geometry

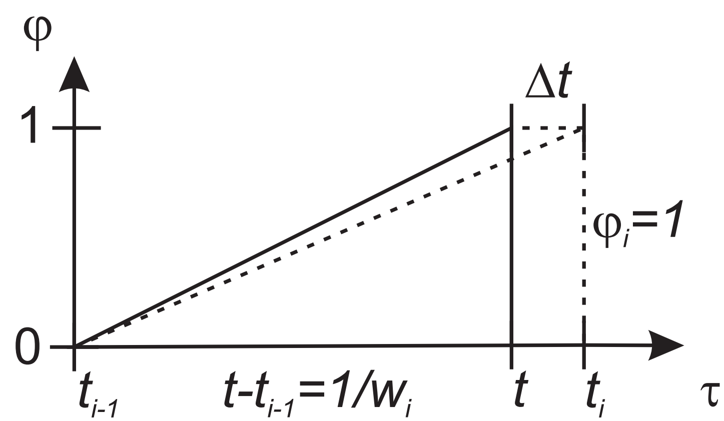
<u>Predicting event too late</u>

$$dw = \alpha y w \frac{(1-\varphi)}{\varphi} d\tau$$



<u>Predicting event too soon</u>

$$dw = -\alpha 1_{\varphi=1} w^2 d\tau$$



Convergence

- Using two different variations of the preceding learning rules, we can show either:
  - *w* = the reward (event *y*) rate under the presence of *x*

$$w_{n+1} = (1-\alpha)w_n + \alpha I_n^{-1}$$

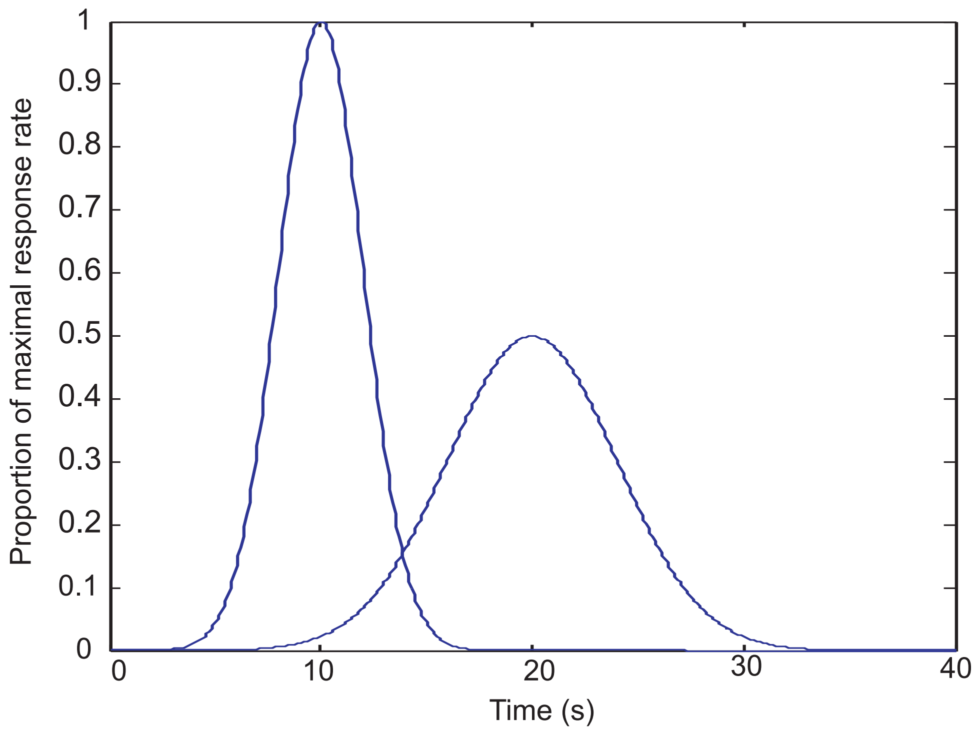
  - *1/w* = the mean time interval between events *y*

$$\frac{1}{w_{i+1}} = (1-\alpha)\frac{1}{w_i} + \alpha I_{i+1}$$

- if 0 < α < 1:
  - The system learns the exponential mobile average
  - The system converges to the mean in a constant number of trials (independent of the time scale)

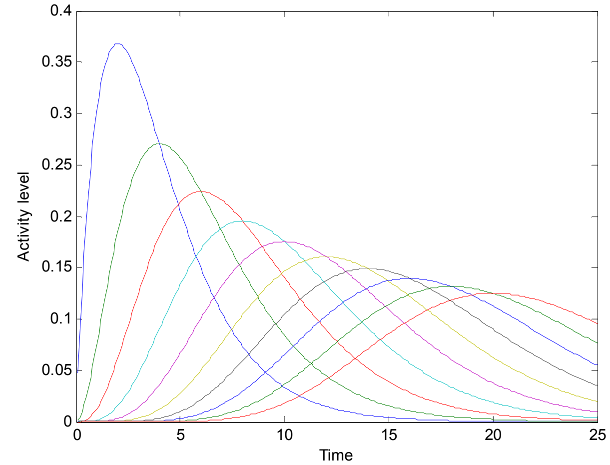
Weber's law

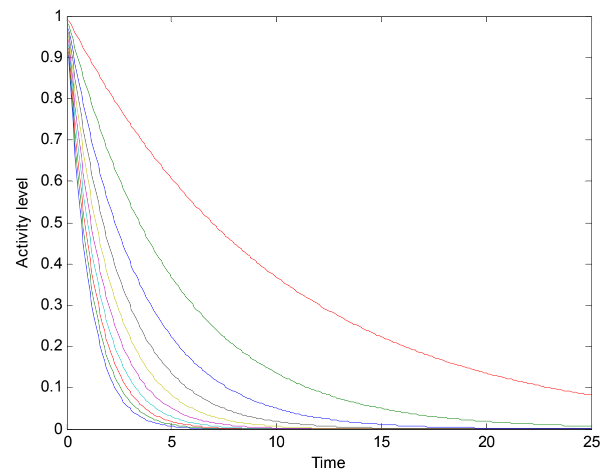
- Timing error is proportional to the time interval

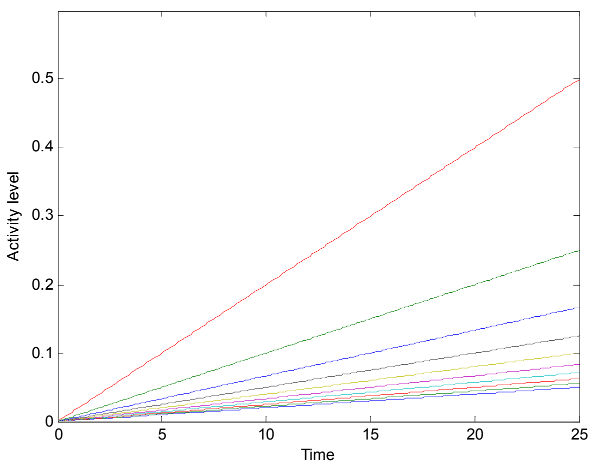


Usual temporal representations

- Diffused delay-lines
- Exponential decay processes
- Linear build-ups (ramps)





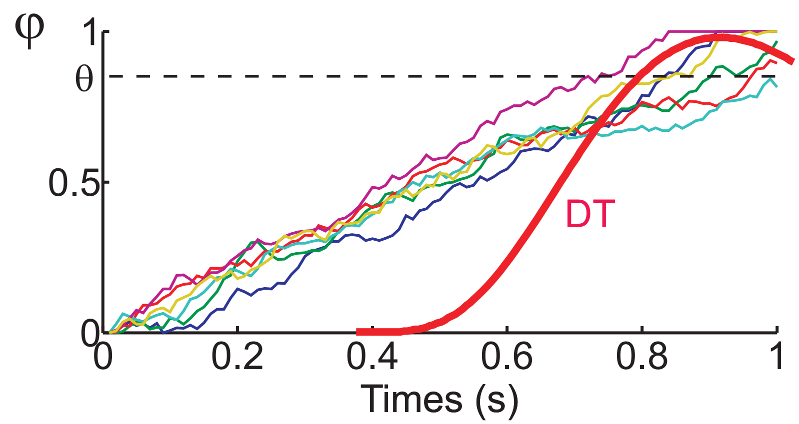


Adding Gaussian noise: The DDP

- Add Gaussian noise *cdW ~ N(0,c²dt)*

$$d\varphi = 1_{\varphi \neq 1} xwd\tau - y\varphi d\tau + cdW$$

- φ is now a drift-diffusion process



- DT: Distribution of time to reach threshold θ

Implementing Weber's law

- For DDP

$$E[DT] = \frac{\theta}{w}\tanh\left(\frac{w\theta}{c^2}\right)$$

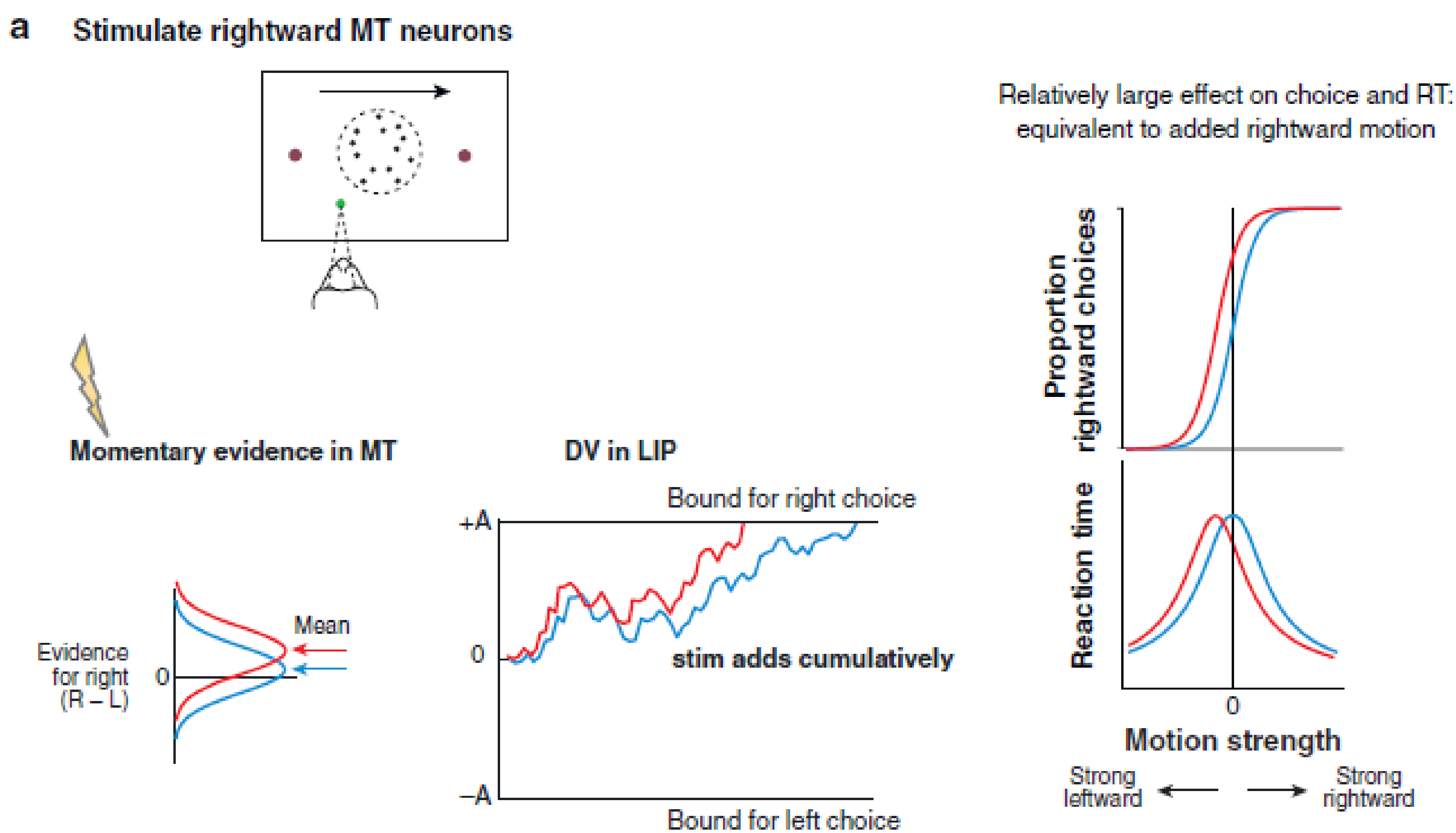
$$Var[DT] = \frac{\theta}{w}\frac{c^2}{w^2}g(y)$$

- Setting $c^2 = \beta^2 w$
- Then $cdW \sim N(0, \beta^2 w d\tau)$
- So we get *E[DT]/STD[DT]* constant (Weber)

Importance of DDP

- DDPs are used to model reaction times
  - Ratcliff & al. 1980’s
- DDPs are use to model decision making (the process of accumulating evidences)
  - Shadlen & al. 2000’s
- BUT: DDPs are usualy fixed in these models,
  - Our work makes DDP timing-adaptive.

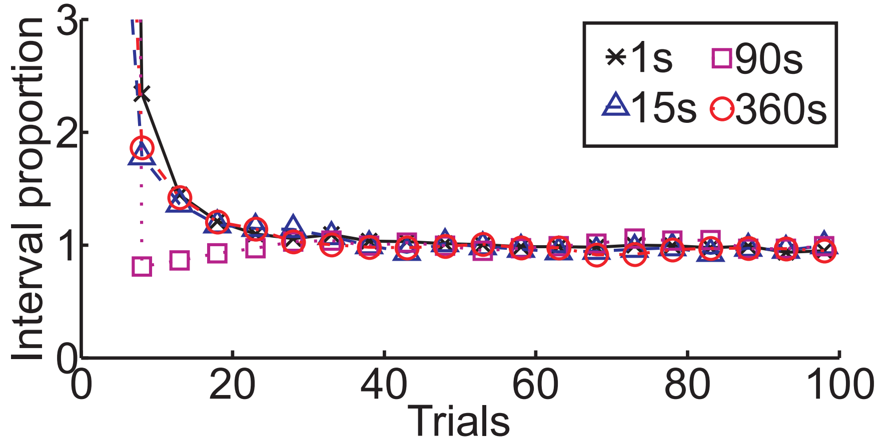
DDP: Accumulation of evidences



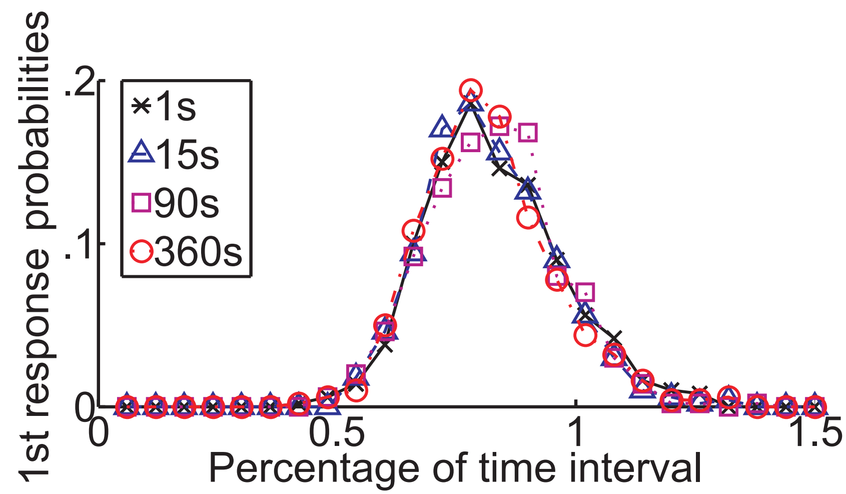
(Gold & Shadlen, 2007)

Results

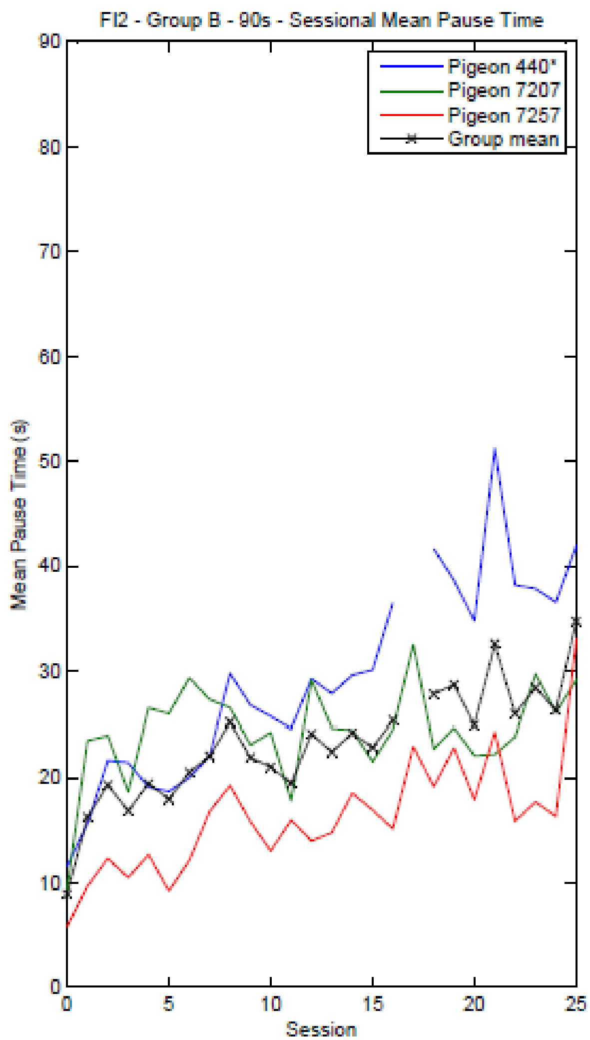
Number of trials independent of time-scale

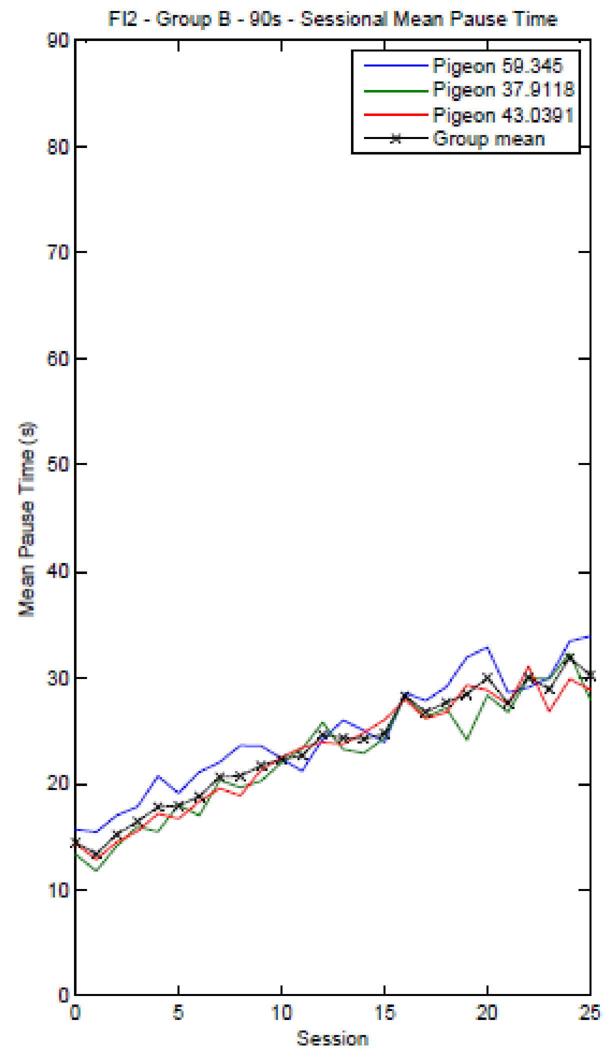


Error proportional to time-scale



Adapting from 30s to 90s

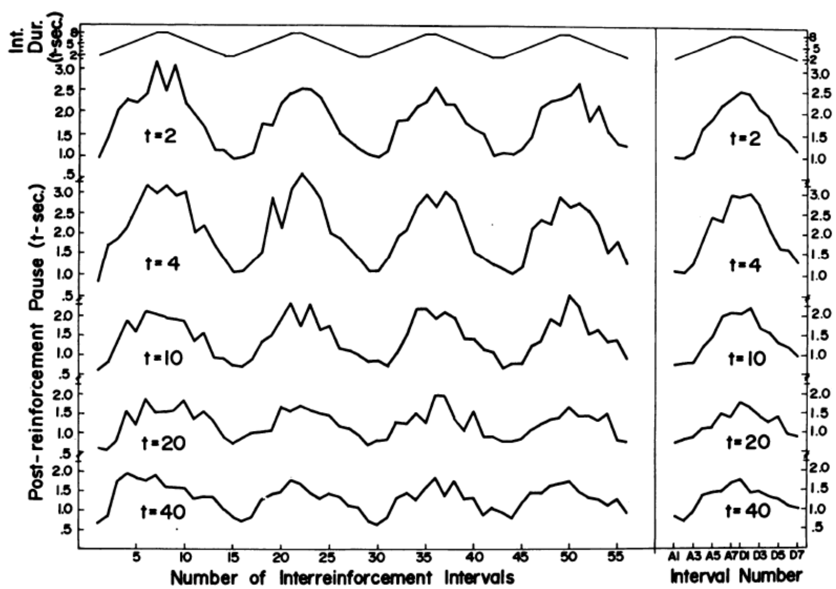




Cyclic schedule tracking results

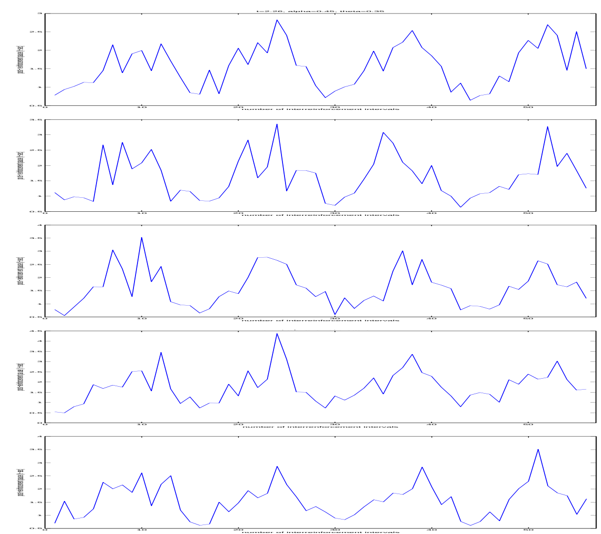
- Intervals kept changing following a cyclic schedule

Animal's data



(Innis & Staddon, 1971)

Adaptive DDP



(Andre Luzardo)

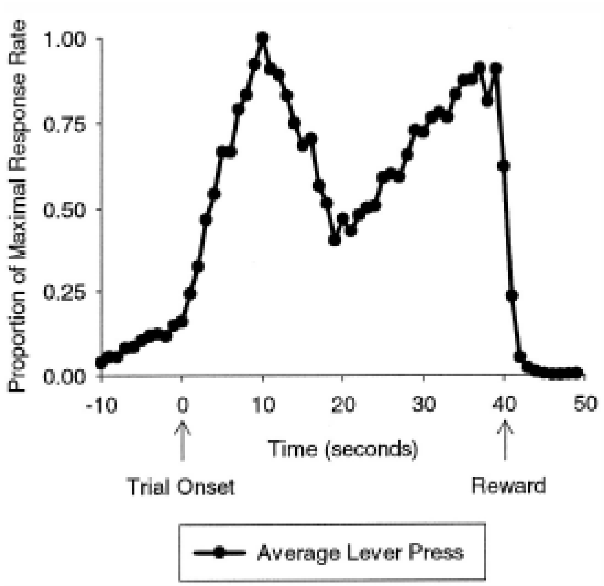
Work in progress…

- Multiple stimuli (multiple inputs)
  - Try to learn the event-rate (or reward-rate) with respect to each stimulus (or input).
  - Eg.: blocking, relative validity, etc.…

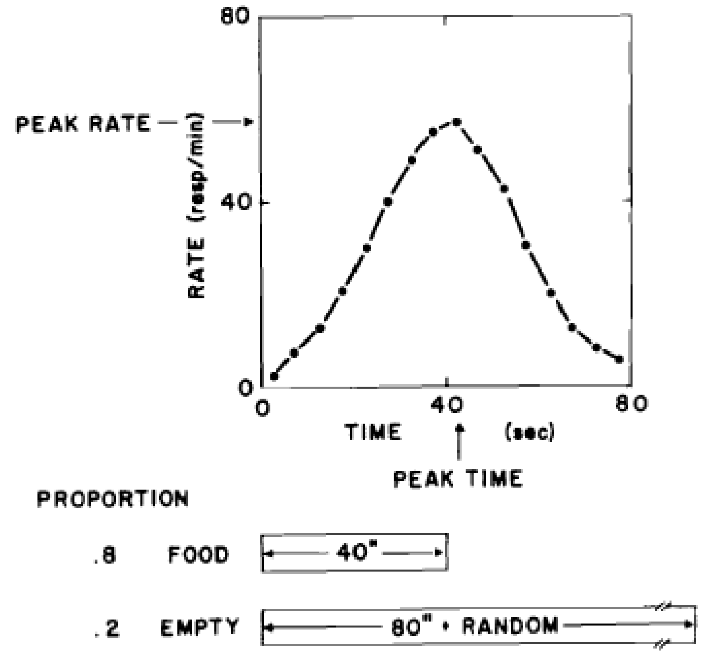
- Multiple intervals (multiple outputs mode)
  - Animal can learn more than one delay for a given event.
  - How to associate the error to the appropriate output?

Future work

- When and how to construct new modes, events, or non-events?



(Matell, Meck & Nicolelis, 2003)



(Roberts, 1981)

Advantages over previous models

- It is the first on-line timing-learning rule for DDP
- More over:
  - It does not required predefined populations of delays.
  - It does not required unbounded integrators variables.
  - The number of trials required to learn is time-scale independent (and works in continuous time).
  - It can track non-stationnary distributions
  - It learns on-line, in real-time.

Conclusion

- This work suggests that accumulation of evidences to make decision and ‘counting’ time (or predicting temporal events) might share the same underlying process.

- Animals are amazing learners compared to most machine learning algorithms.
- The animal literature is still filled with amazing facts and hints about how they do it.

Acknowledgements

- Organisers:
  - Rich Sutton
  - Doina Precup
  - Elliot Ludvig
- Collaborators:
  - Yoshua Bengio (UMontréal)
  - Elliot Ludvig (UPrinceton)
  - Andre Luzardo (USaoPaulo)
- Funding:

Natural Sciences and Engineering Research Council of Canada — Conseil de recherches en sciences naturelles et en génie du Canada

Canadian Institutes of Health Research — Instituts de recherche en santé du Canada

Défense nationale — National Defence

TRUTH • DUTY • VALOUR
VÉRITÉ • DEVOIR • VAILLANCE

- Paper available on arXiv:
  - *http://arxiv.org/abs/1103.2382*

francois.rivest@rmc.ca