A systems factorial technology approach to classifying the architecture of fraction perception

Thomas J. Faulkenberry

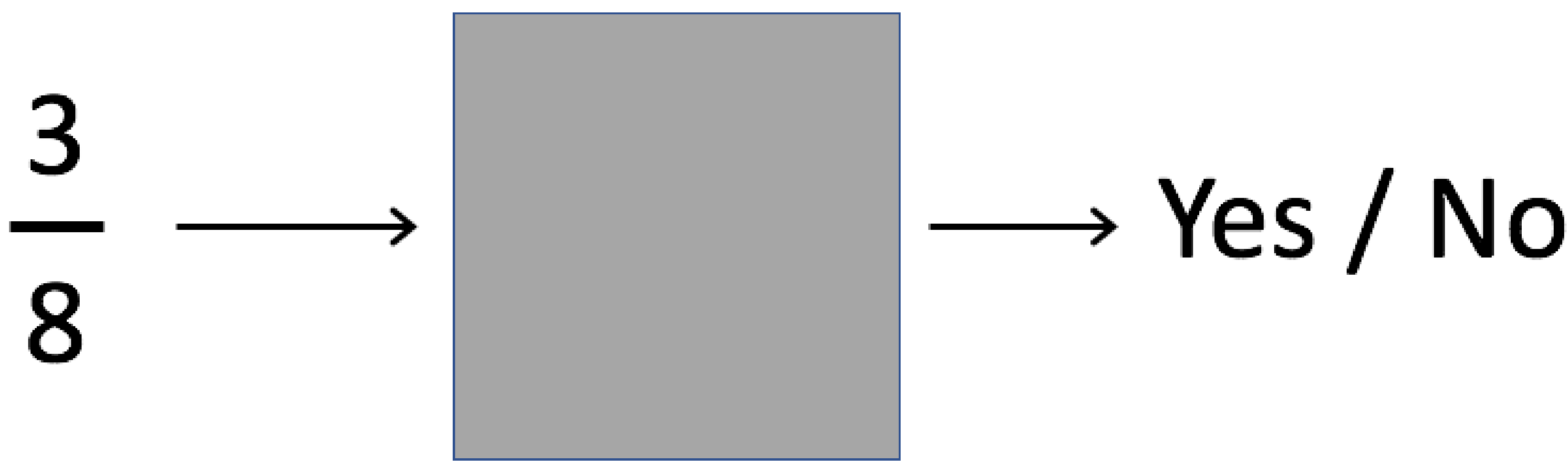
(joint work with Sabrina Hetzel & Kristen Bowman)

Tarleton State University

SWPA 2019

Task: decide if fraction contains a number greater than 5 in *either* component.

Question: how do we make this decision?



Imagine our mental "factory" has two workers, Nan and Dennis, responsible for making the decision for the numerator and denominator, respectively.

Imagine our mental "factory" has two workers, Nan and Dennis, responsible for making the decision for the numerator and denominator, respectively.

Some possibilities:

- Nan looks at numerator first, then passes the fraction to Dennis, who looks at denominator (regardless of Nan's decision)

Imagine our mental "factory" has two workers, Nan and Dennis, responsible for making the decision for the numerator and denominator, respectively.

Some possibilities:

- Nan looks at numerator first, then passes the fraction to Dennis, who looks at denominator (regardless of Nan's decision)

- Nan looks at numerator first, only passing to Dennis if her component doesn't satisfy "greater than 5" condition.

Imagine our mental “factory” has two workers, Nan and Dennis, responsible for making the decision for the numerator and denominator, respectively.

Some possibilities:

- Nan looks at numerator first, then passes the fraction to Dennis, who looks at denominator (regardless of Nan's decision)

- Nan looks at numerator first, only passing to Dennis if her component doesn't satisfy “greater than 5” condition.

- Nan and Dennis look at their components at the same time, and fraction is passed on once both Nan and Dennis have made their respective decisions

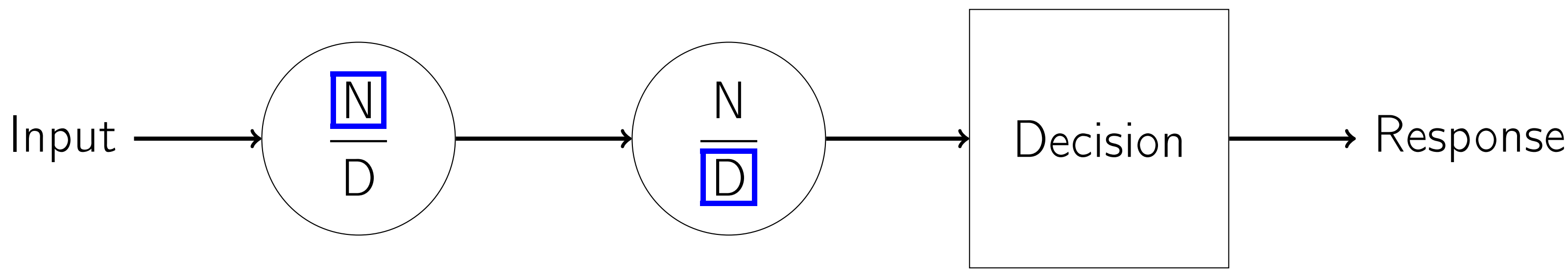
Imagine our mental "factory" has two workers, Nan and Dennis, responsible for making the decision for the numerator and denominator, respectively.

Some possibilities:

- Nan looks at numerator first, then passes the fraction to Dennis, who looks at denominator (regardless of Nan's decision)
- Nan looks at numerator first, only passing to Dennis if her component doesn't satisfy "greater than 5" condition.
- Nan and Dennis look at their components at the same time, and fraction is passed on once both Nan and Dennis have made their respective decisions
- Nan and Dennis look at their components at the same time. If one of them finds that their component satisfies "greater than 5" condition, the fraction is immediately passed on.

Serial architecture

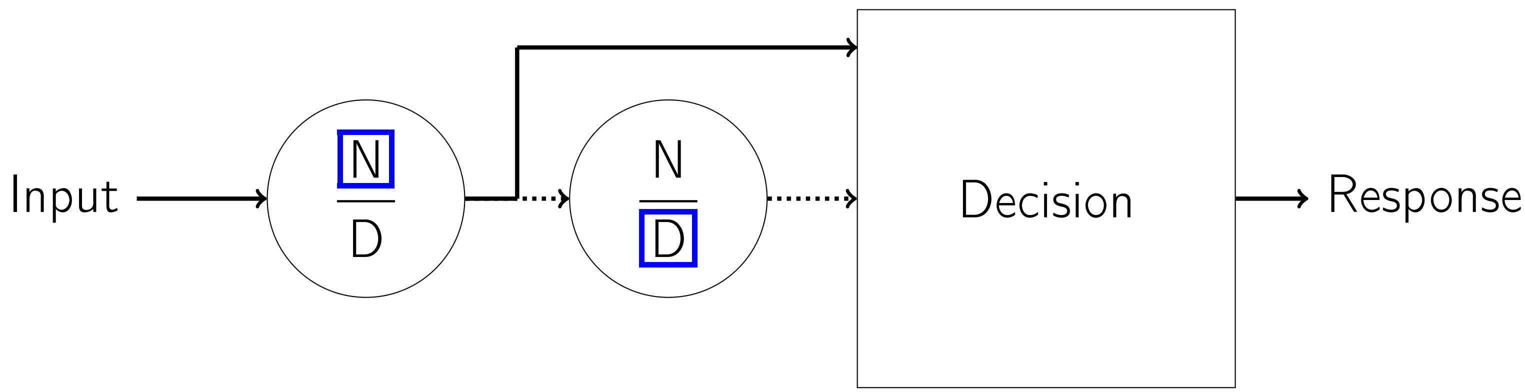
Stopping rule = exhaustive



Each target is processed *sequentially* – both N and D must complete before response is made

Serial architecture

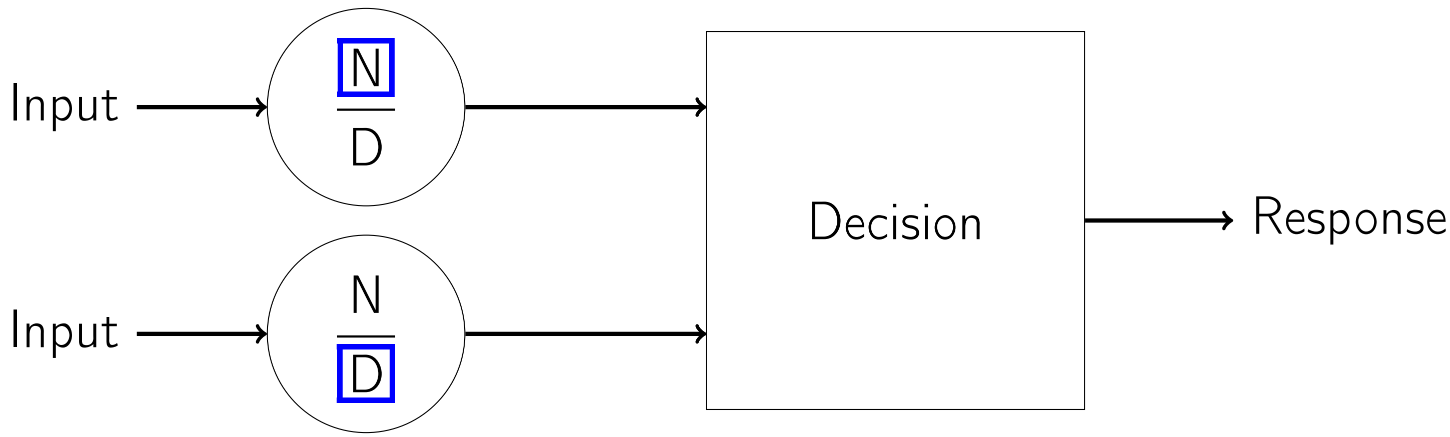
Stopping rule = self-terminating



Each target is processed *sequentially* – but either N or D is sufficient to trigger response

Parallel architecture

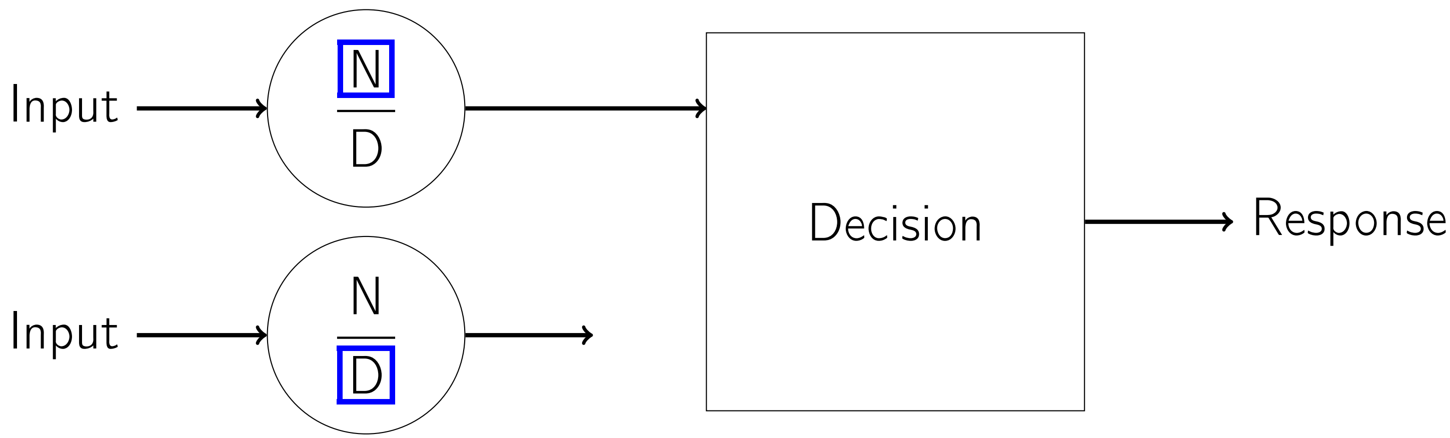
Stopping rule = exhaustive



Each target is processed *simultaneously* – both A and B must complete before response is made

Parallel architecture

Stopping rule = self-terminating



Each target is processed *simultaneously* – but either A or B is sufficient to trigger response

Our goal is to determine which of these architectures governs how we process fractions.

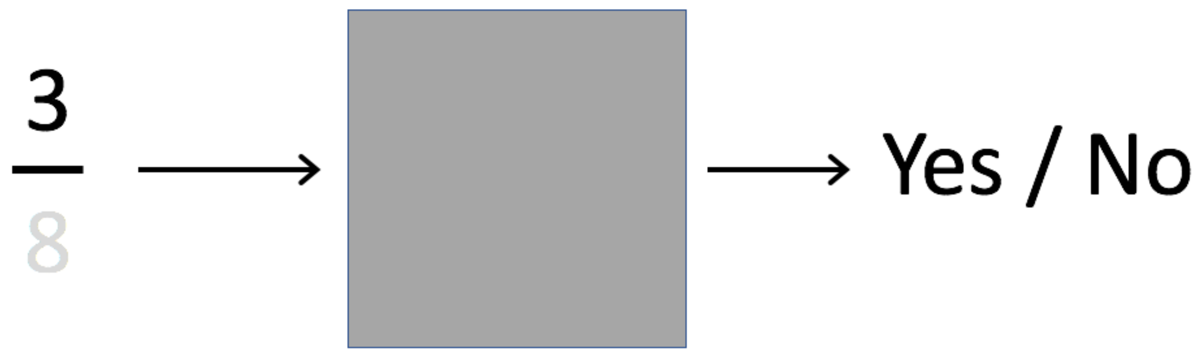
Our goal is to determine which of these architectures governs how we process fractions.

Unfortunately, we cannot *directly* observe how our "workers" Nan and Dennis handle their respective tasks.

Our goal is to determine which of these architectures governs how we process fractions.

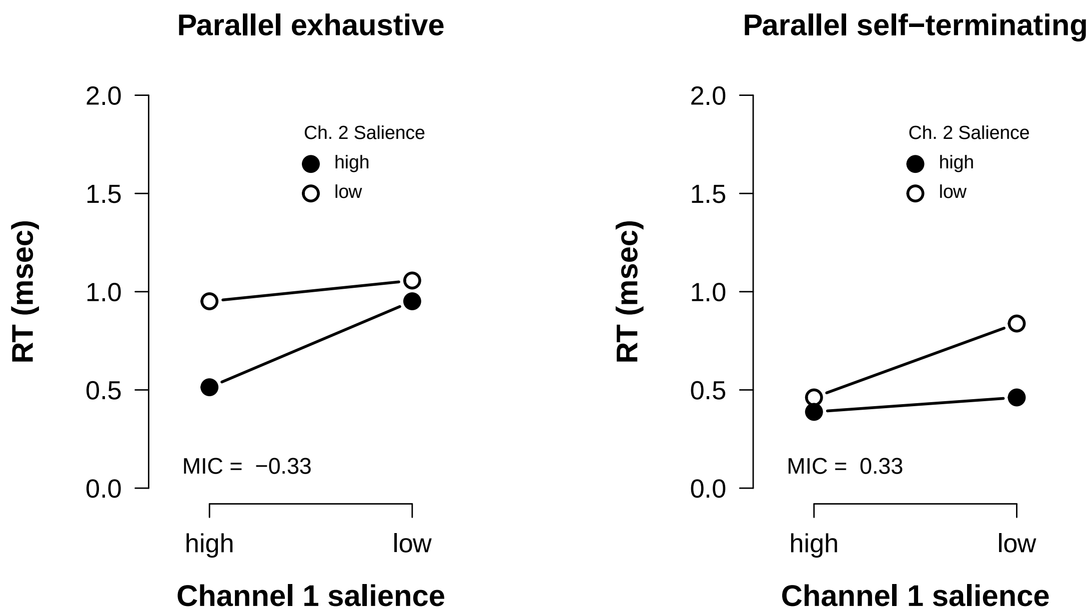
Unfortunately, we cannot *directly* observe how our "workers" Nan and Dennis handle their respective tasks.

However, we can *indirectly* observe them by manipulating the inputs they receive and measuring the effect on performance.

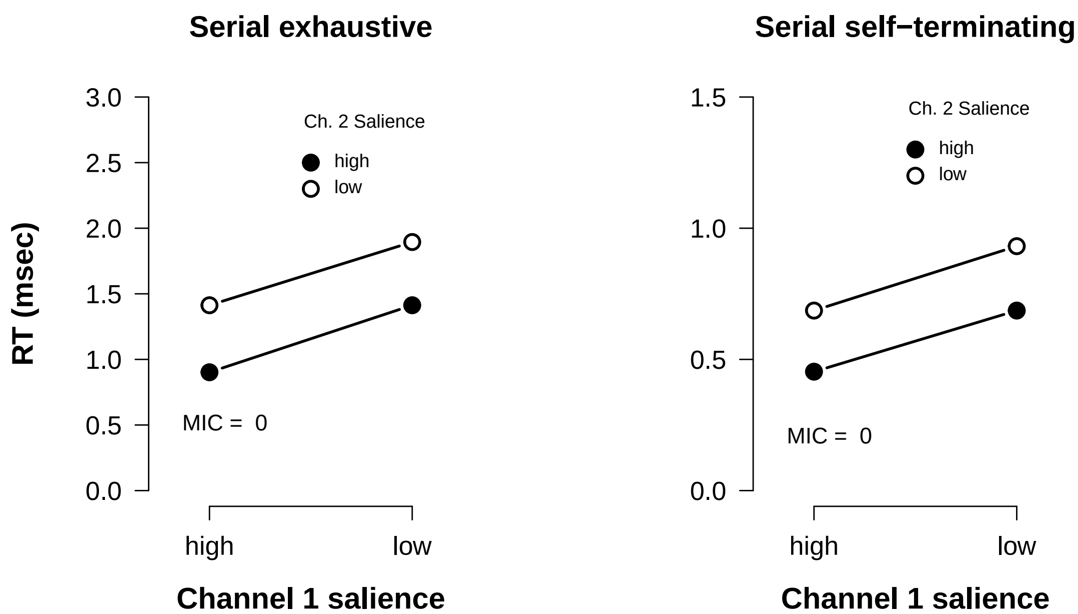


We call this a **salience** manipulation. The goal is to make the task harder by manipulating how easy it is for Nan/Dennis to make their decisions.

Classically, problems of this type have been studied by comparing **mean** response times (RTs) in each of the salience conditions.



Classically, problems of this type have been studied by comparing **mean** response times (RTs) in each of the salience conditions.

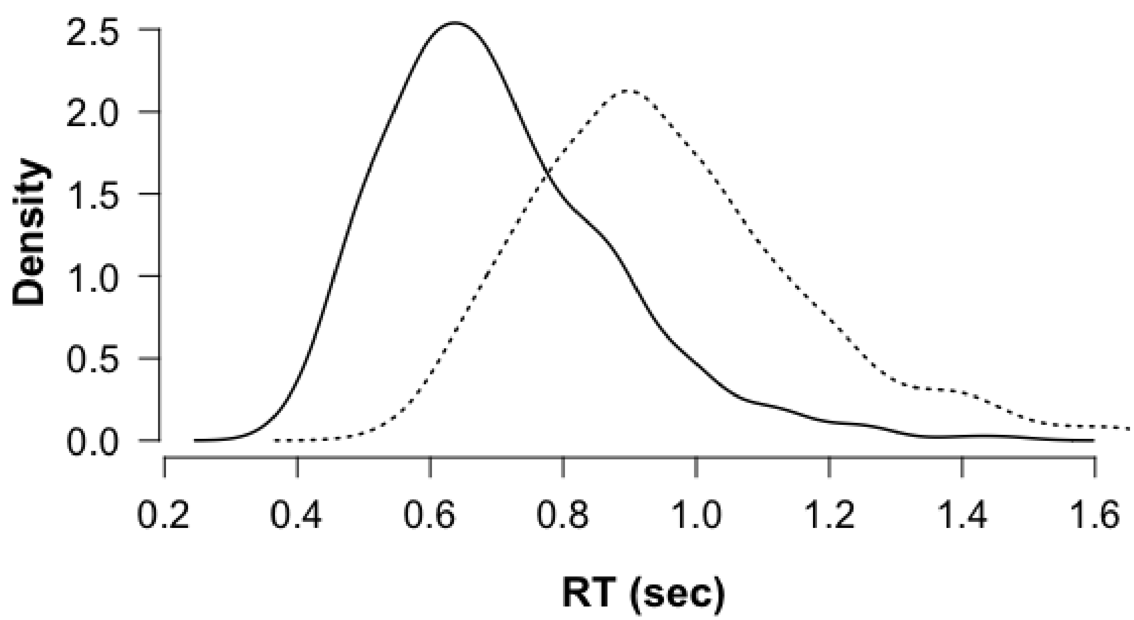


Problem – model mimicry: these techniques cannot distinguish between different stopping rules for serial architecture.

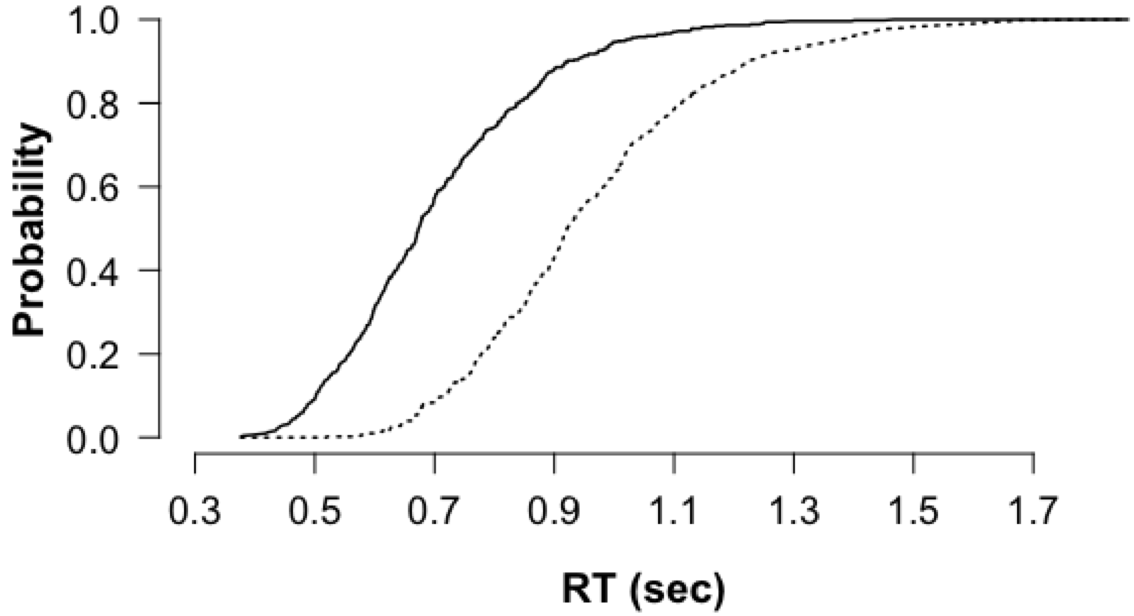
A way out of this problem is to use the entire *distribution* of RTs.

Some analytic tools

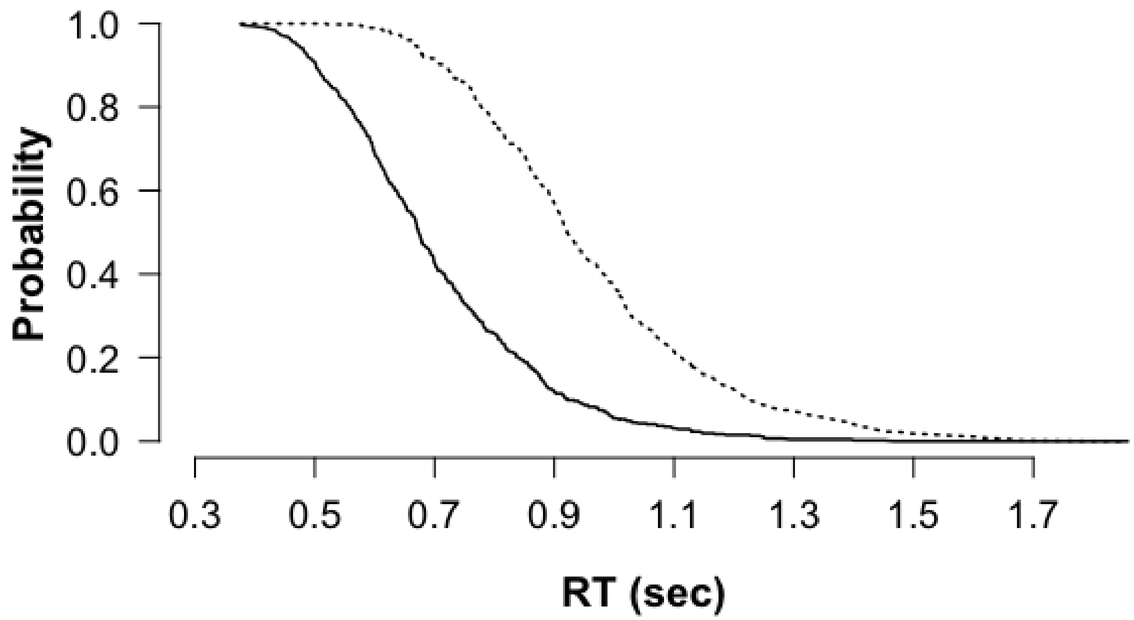
Probability density function: $f_X(x)$



Cumulative distribution function: $F_X(x) = \int_{-\infty}^{x} f_X(t)dt$



Survivor function: $S_X(x) = \int_x^\infty f_X(t) = 1 - F_X(x)$

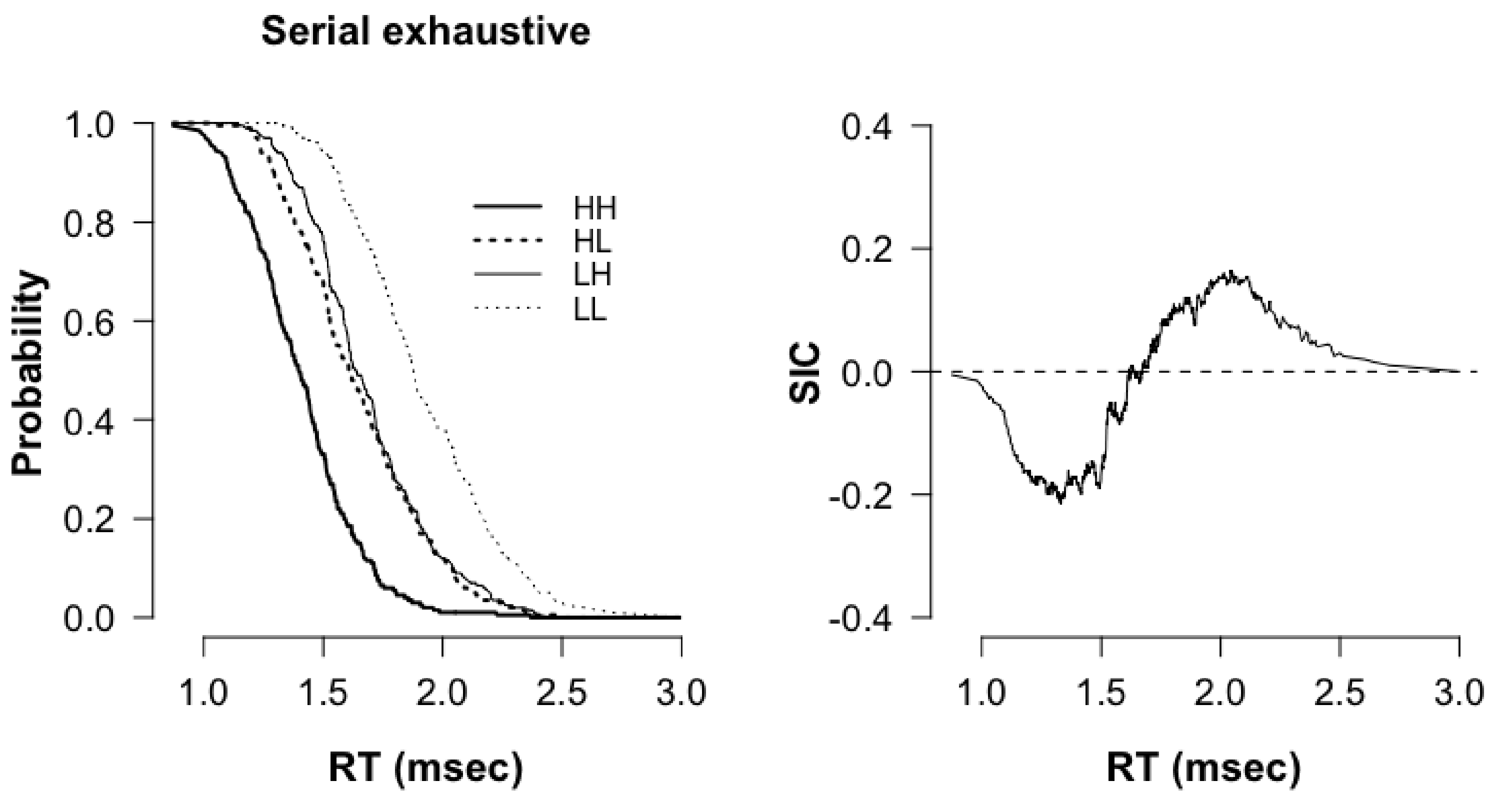


In an experiment, we collect distributions of RTs from 4 different salience conditions: HH, HL, LH, and LL.

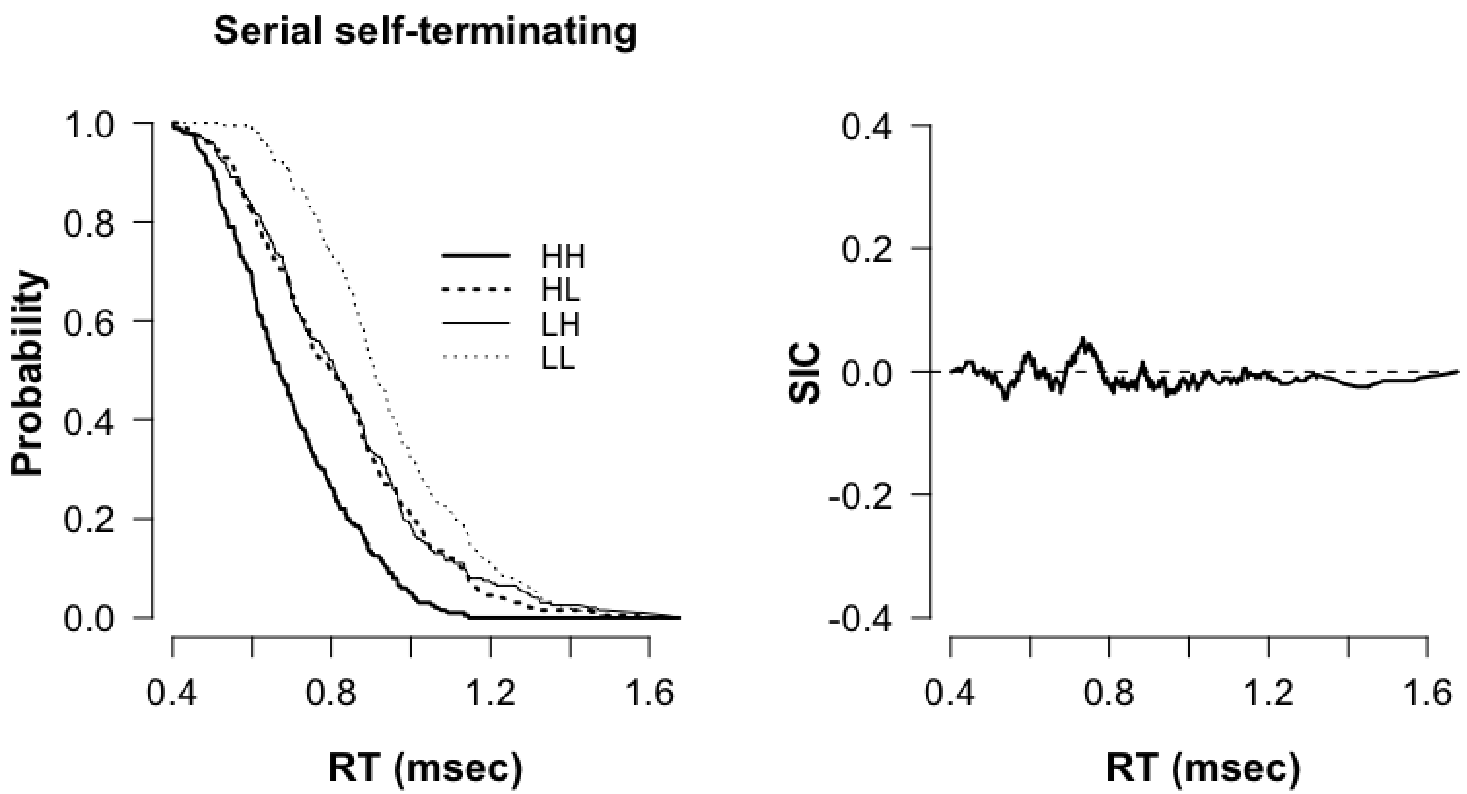
If we fit a survivor curve to each distribution, we can generate the **survivor interaction contrast**, or **SIC**, as follows:

$$SIC = \Big(S_{LL} - S_{LH}\Big) - \Big(S_{HL} - S_{HH}\Big)$$

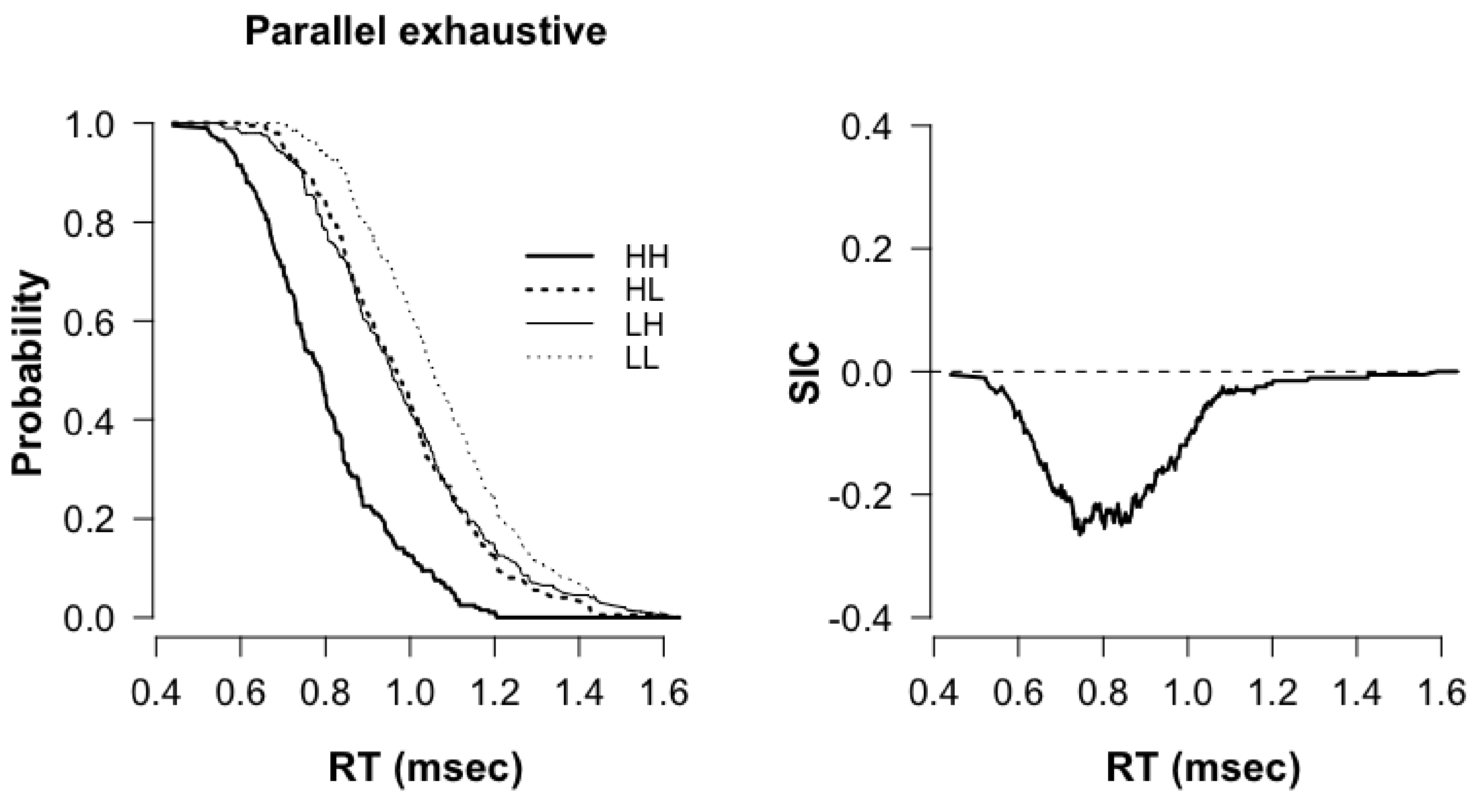
It turns out that this function is *really* useful (Townsend & Nozawa, 1995).



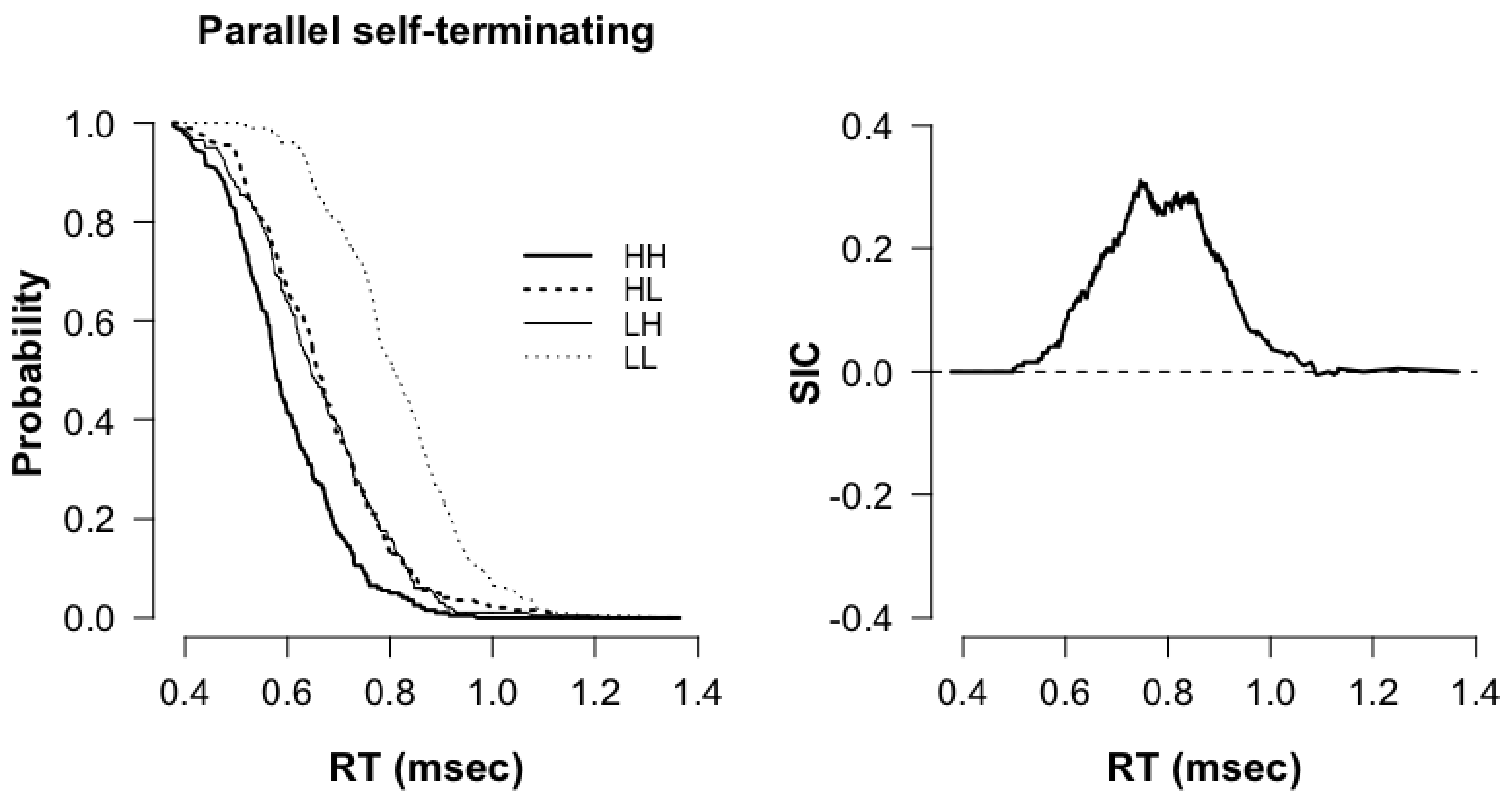
Theorem 1. *For a serial exhaustive model, there exists a point* t_0 *such that for* $t < t_0$*,* $SIC(t) < 0$*, and for* $t > t_0$*,* $SIC(t) > 0$.



Theorem 2. *For a serial self-terminating model,* $SIC = 0$.



Theorem 3. *For a parallel, exhaustive model,* $SIC(t) < 0$ *for all* $t > 0$.



Theorem 4. *For a parallel, self-terminating model,* $SIC(t) > 0$ *for all* $t > 0$.

Task: decide if fraction contains a number greater than 5 in *either* component.

Stimuli:

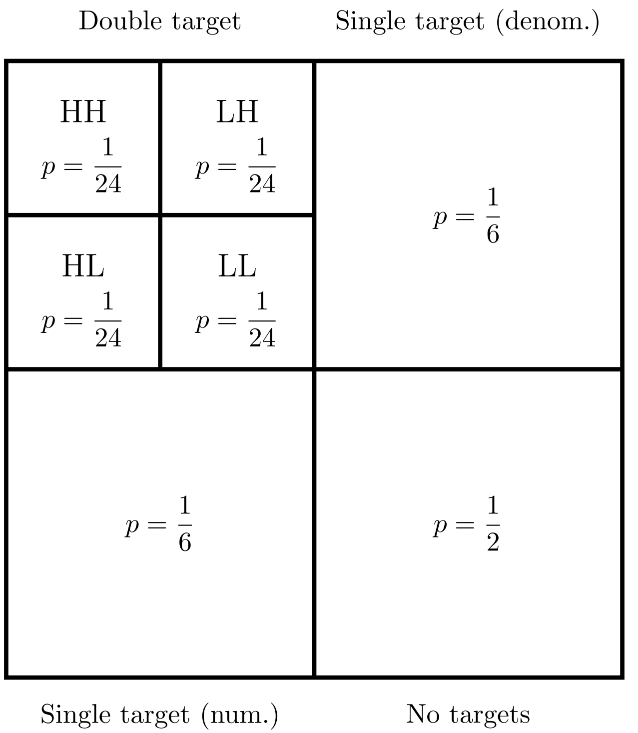
- numerators = 2,3,4,6,7,8
- denominators = 2,3,4,6,7,8
- 36 possible fractions
- how many times do we repeat them?

		Numerator			
		greater than 5		less than 5	
		Salience: Numerator		*Salience: Numerator*	
Denominator	*Salience: Denominator*	*high*	*low*	*high*	*low*
greater than 5	*high*	6 - 7	6 - 7	2 - 7	2 - 7
greater than 5	*low*	6 - 7	6 - 7	2 - 7	2 - 7
less than 5	*high*	6 - 2	6 - 2	2 - 3	2 - 3
less than 5	*low*	6 - 2	6 - 2	2 - 3	2 - 3

Task: decide if fraction contains a number greater than 5 in *either* component.

Stimuli:

- 36 fractions
- need between 100 and 200 trials in each double target condition
- 5 reps of 36 = 180 trials
- 180 trials = 1/24 of stimulus set
- $24 \times 180 = 4,320$ trials

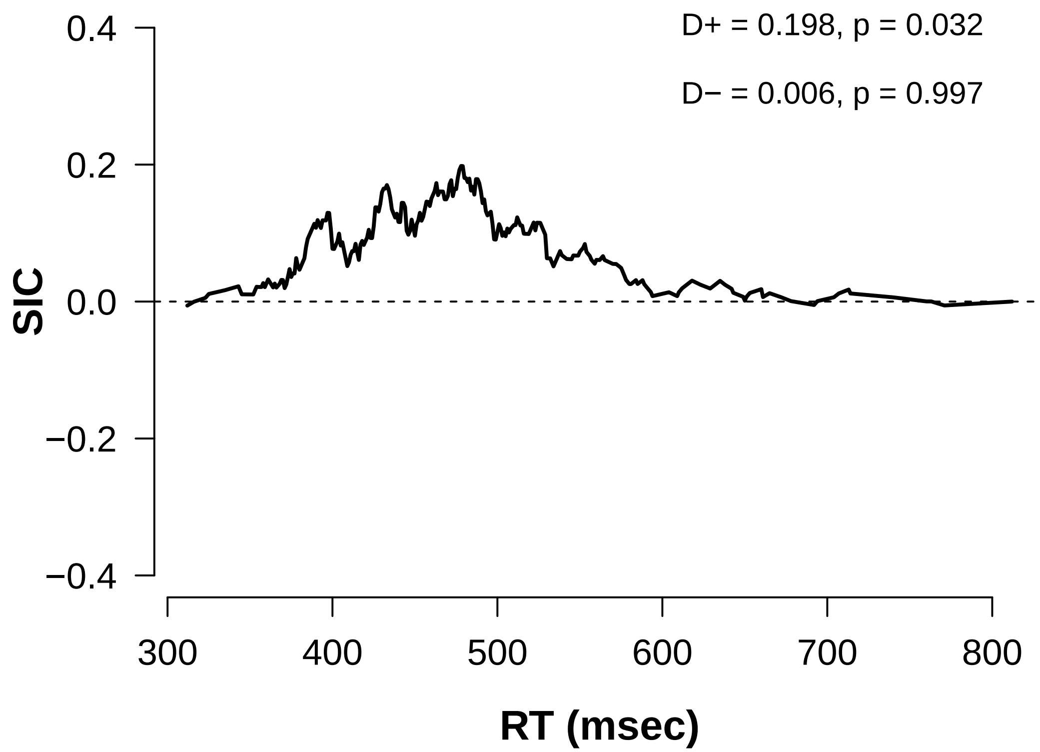


Analytic workflow: for each subject, we:

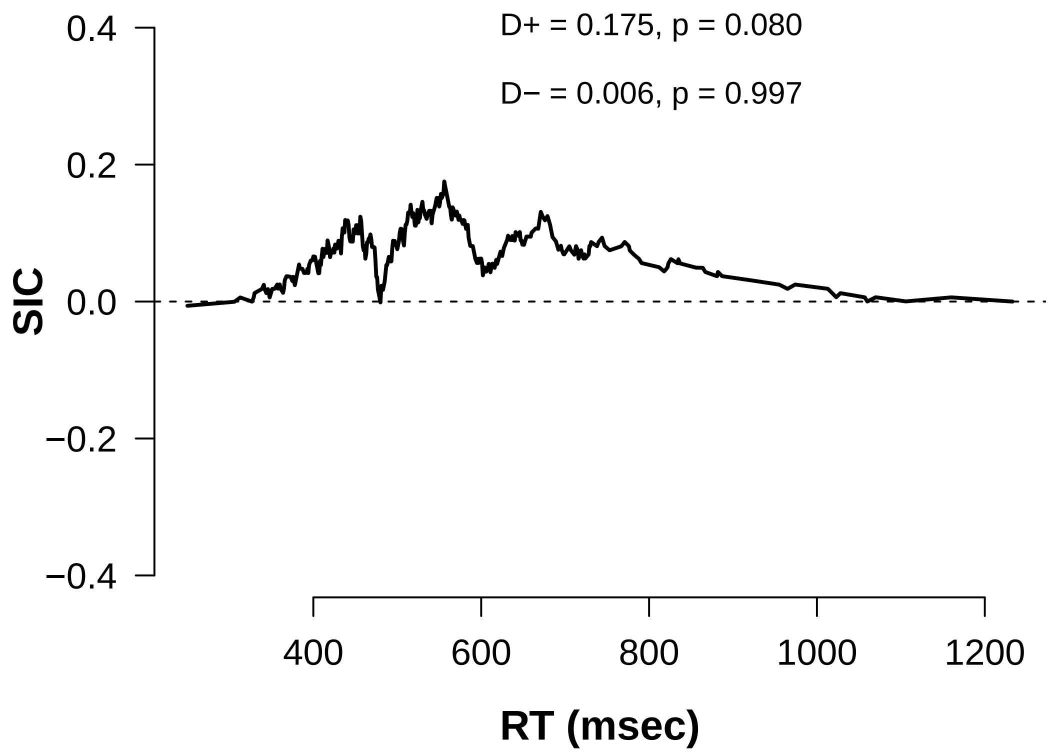
- filter out errors ($M = 3.75\%$) and RT outliers ($M = 1.6\%$)
- estimate survivor functions for each double-target condition
- plot and visually inspect survivor interaction contrast (SIC)
- use Houpt-Townsend statistic[1] to test whether SIC is positive and/or negative

[1] Houpt, J. W., & Townsend, J. T. (2010). The statistical properties of the survivor interaction contrast. *Journal of Mathematical Psychology, 54*, 446-453.

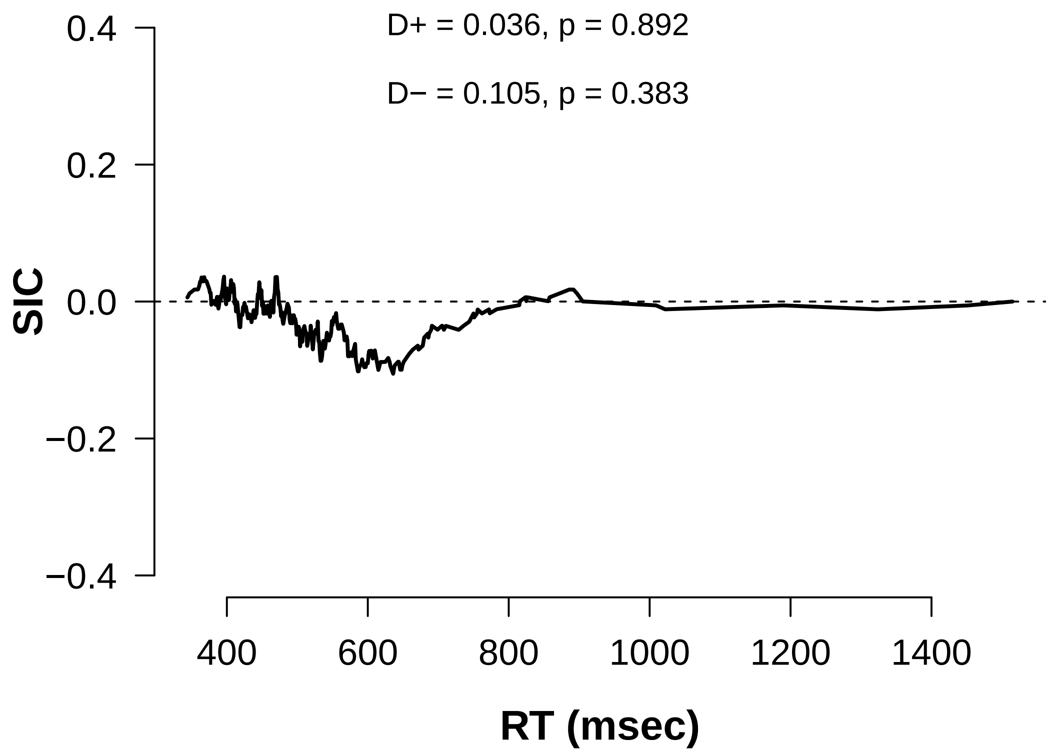
Subject 1:


D+ = 0.198, p = 0.032
D− = 0.006, p = 0.997
0.4
0.2
0.0
−0.2
−0.4
SIC
300
400
500
600
700
800
RT (msec)

Subject 2:

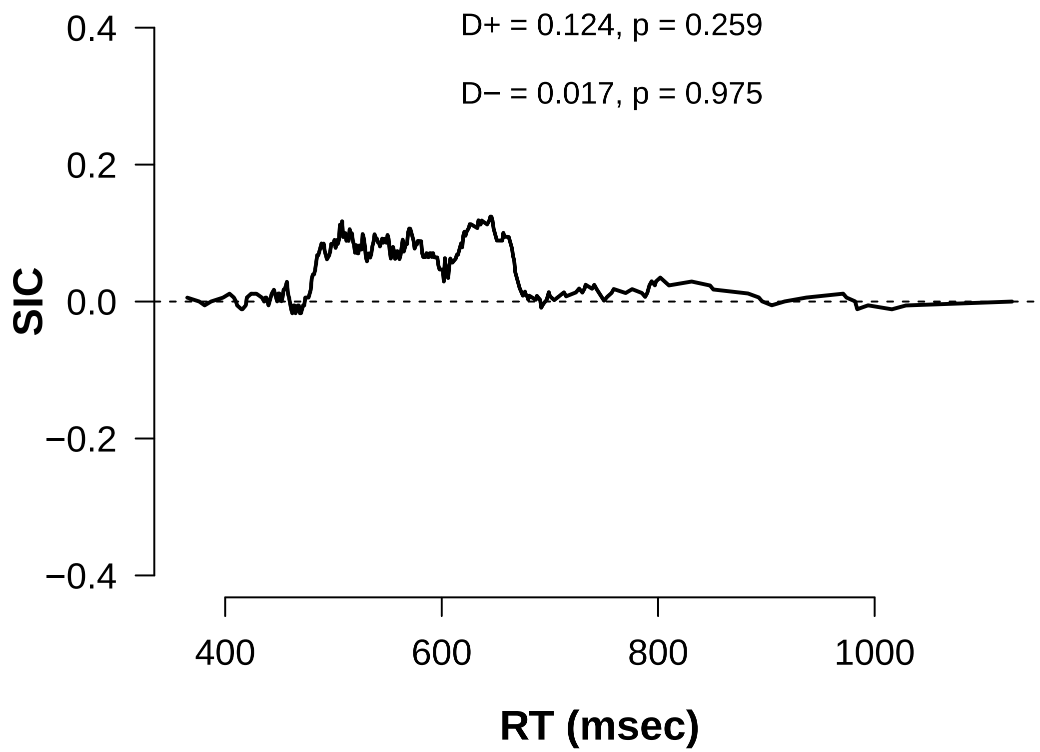

D+ = 0.175, p = 0.080
D− = 0.006, p = 0.997
SIC
0.4
0.2
0.0
−0.2
−0.4
400
600
800
1000
1200
RT (msec)

Subject 3:

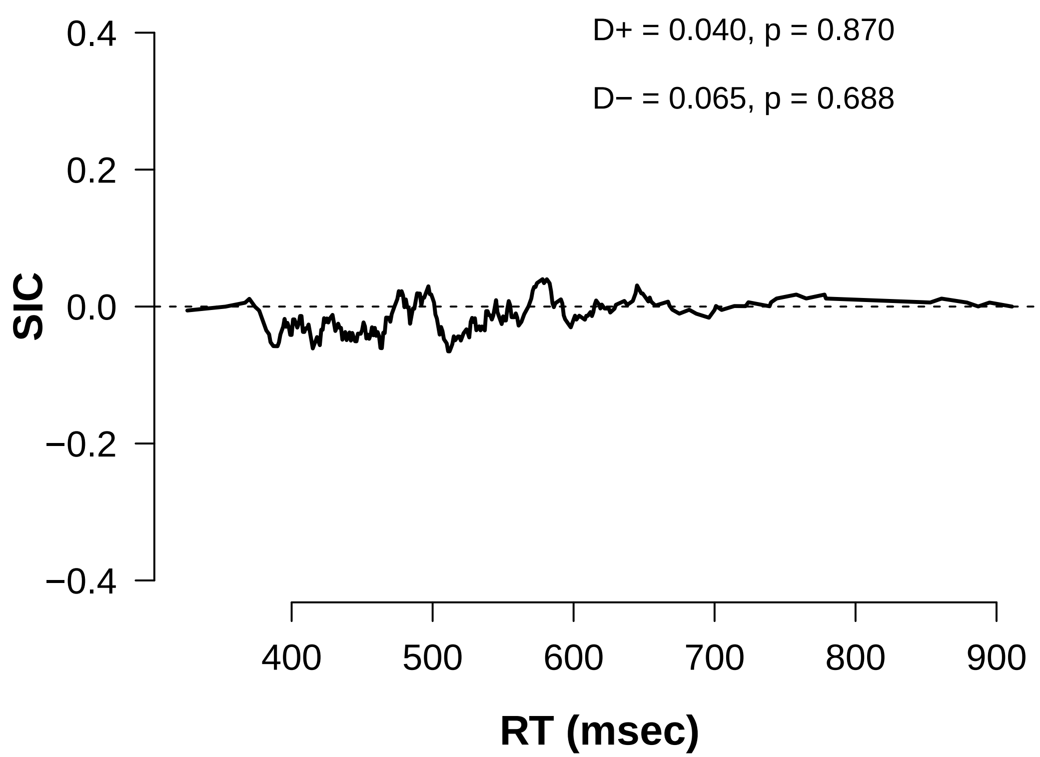


D+ = 0.036, p = 0.892
D− = 0.105, p = 0.383
0.4
0.2
0.0
−0.2
−0.4
SIC
400
600
800
1000
1200
1400
RT (msec)

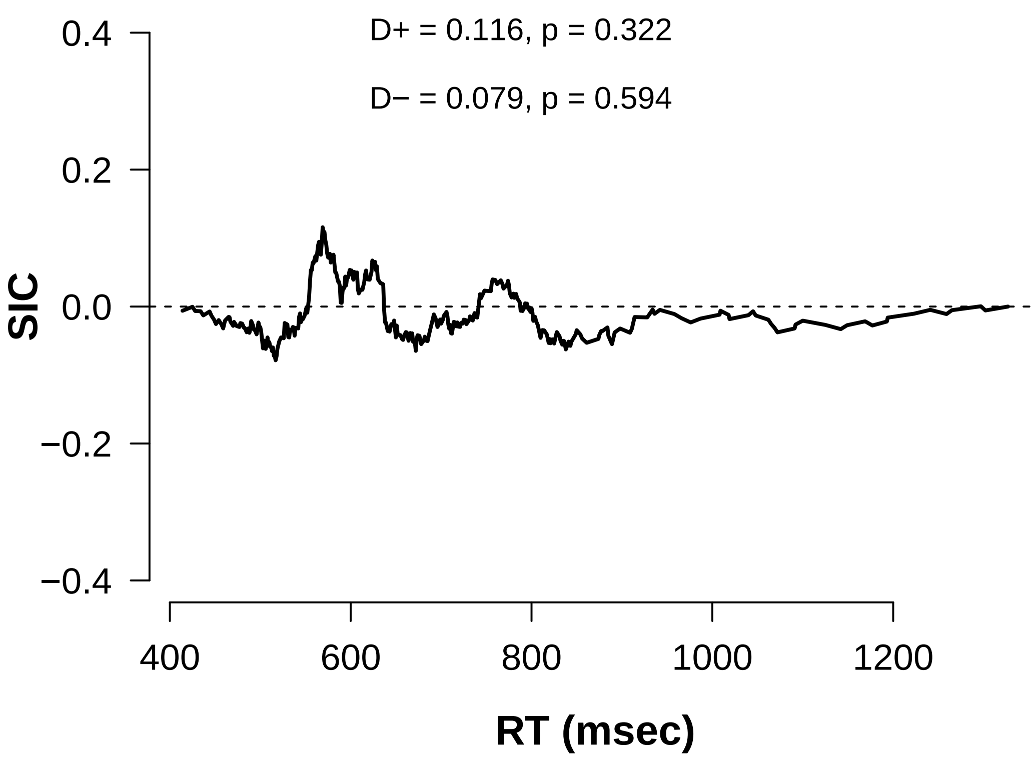
Subject 4:


D+ = 0.124, p = 0.259
D− = 0.017, p = 0.975
0.4
0.2
0.0
−0.2
−0.4
SIC
400
600
800
1000
RT (msec)

Subject 5:

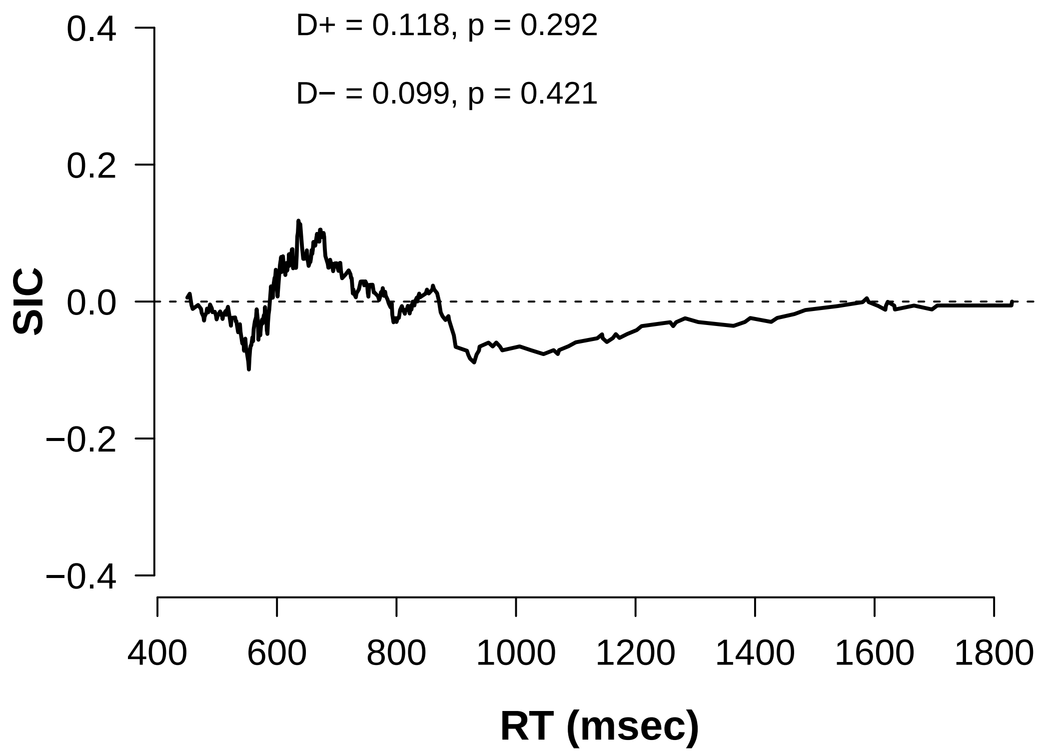

D+ = 0.040, p = 0.870
D− = 0.065, p = 0.688
0.4
0.2
0.0
−0.2
−0.4
SIC
400
500
600
700
800
900
RT (msec)

Subject 6:



D+ = 0.116, p = 0.322
D− = 0.079, p = 0.594
0.4
0.2
0.0
−0.2
−0.4
SIC
400
600
800
1000
1200
RT (msec)

Subject 7:


D+ = 0.118, p = 0.292
D− = 0.099, p = 0.421
0.4
0.2
0.0
−0.2
−0.4
SIC
400
600
800
1000
1200
1400
1600
1800
RT (msec)

Conclusions:

- SIC indicated **serial self-terminating** architecture for 6 of 7 subjects
- capacity functions[2] all indicate serial self-terminating (even for subj 1)
- people process fraction components in a serial fashion.

[2] capacity analysis involves using cumulative hazard functions to analyze how adding information on single processing channels affects the processing of the system as a whole. See Houpt and Townsend (2012).

Thank you!

- Thanks to Tarleton Office of Research and Innovation for funding!
- slides available at `github.com/tomfaulkenberry/talks`
- Twitter: `@tomfaulkenberry`
- Email: `faulkenberry@tarleton.edu`

Capacity analysis

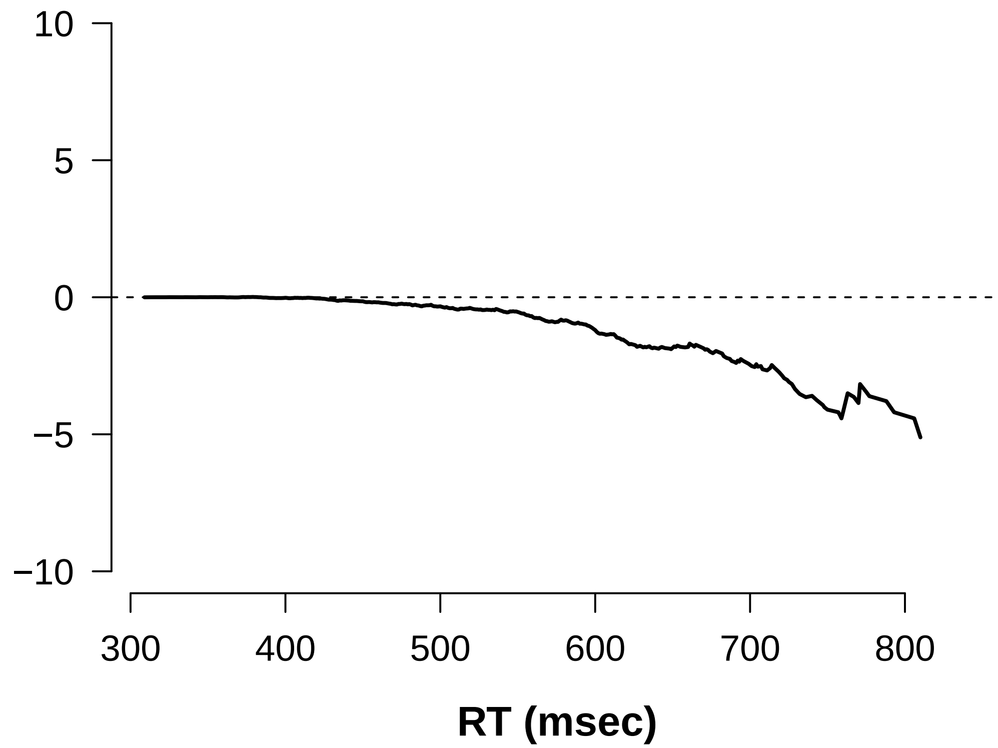
Workflow: for each subject, we:

- collect RTs for both single-target conditions (A and B) and the double-target condition (AB)
- compute and plot capacity function

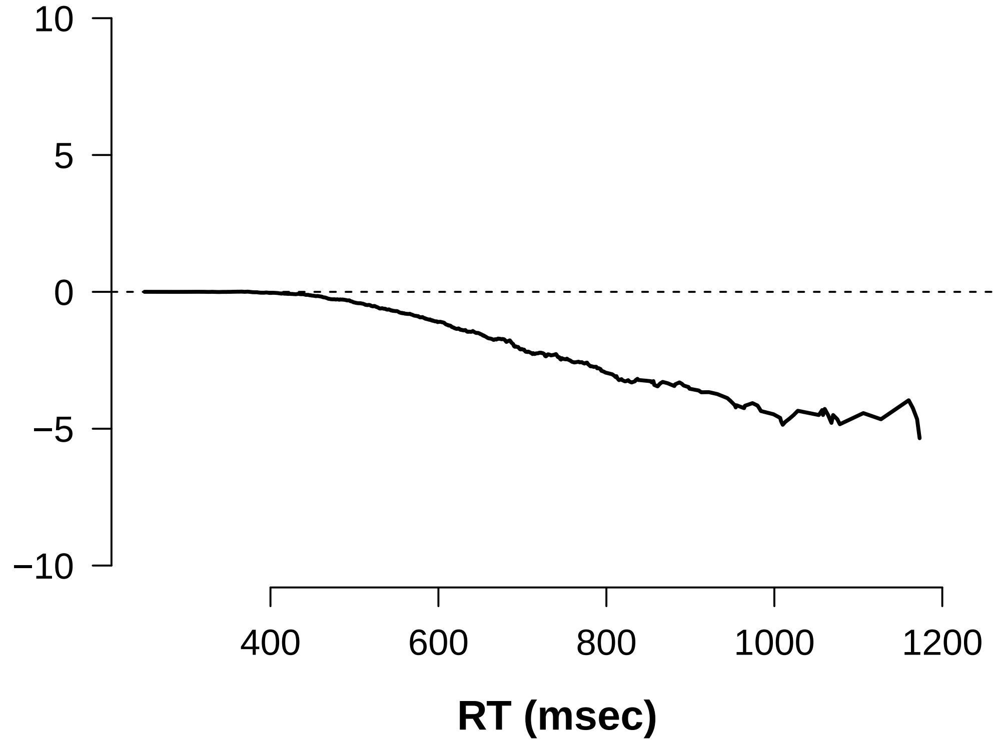
$$C_{\text{OR}}(t) = H_{AB}(t) - (H_A(t) + H_B(t))$$

- plot and visually inspect capacity function

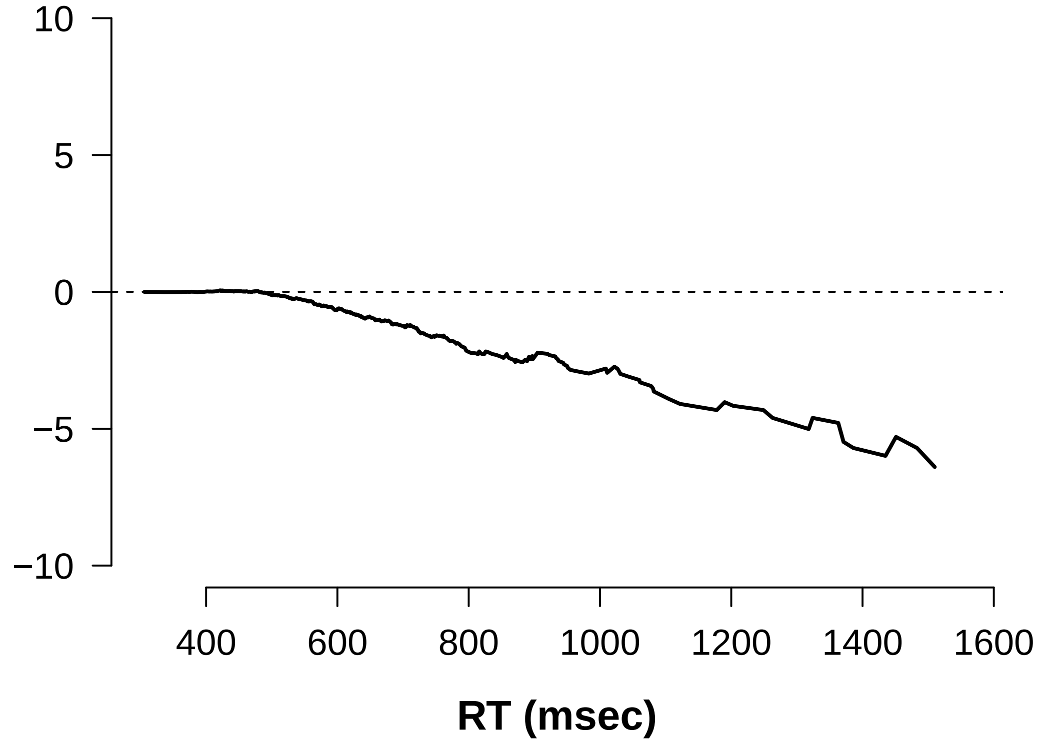
Subject 1



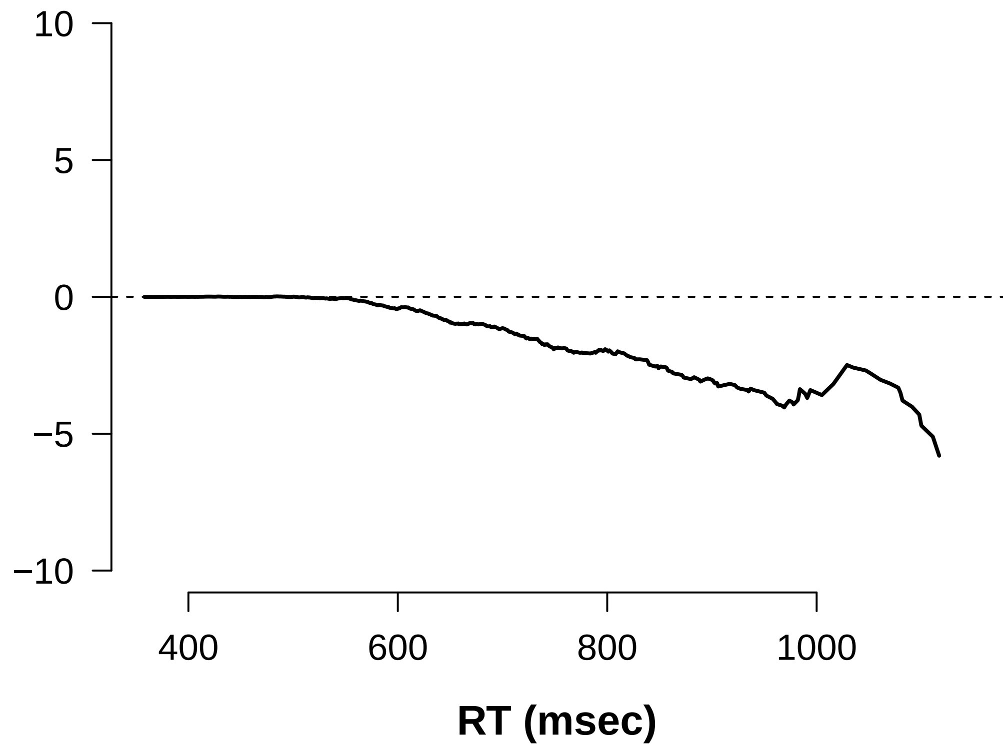
Subject 2



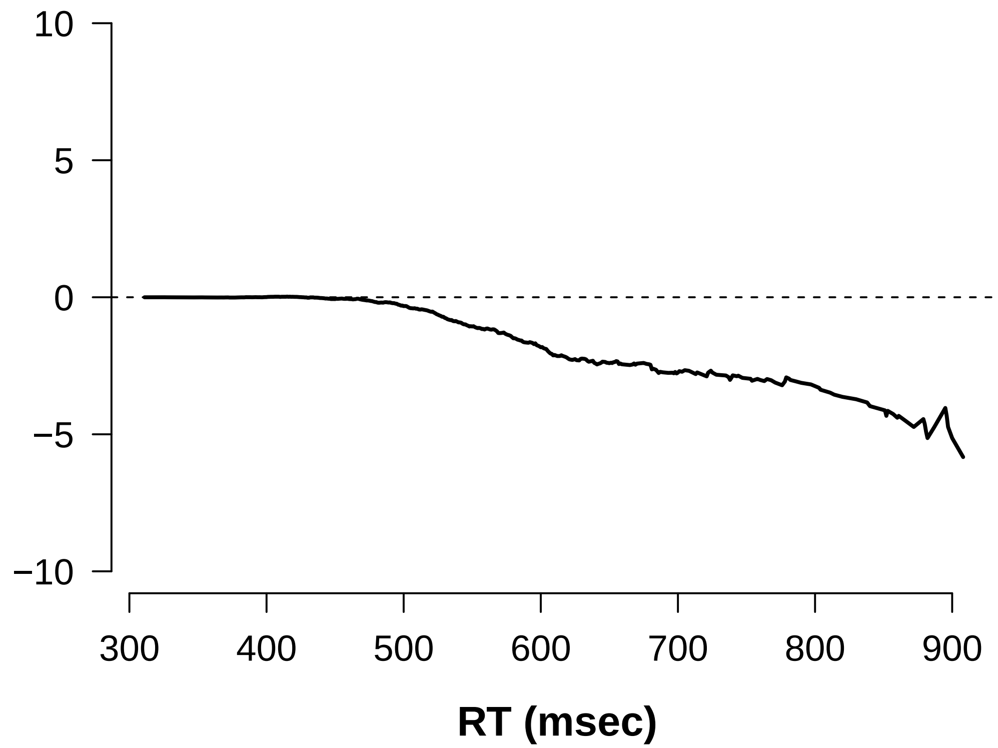
Subject 3



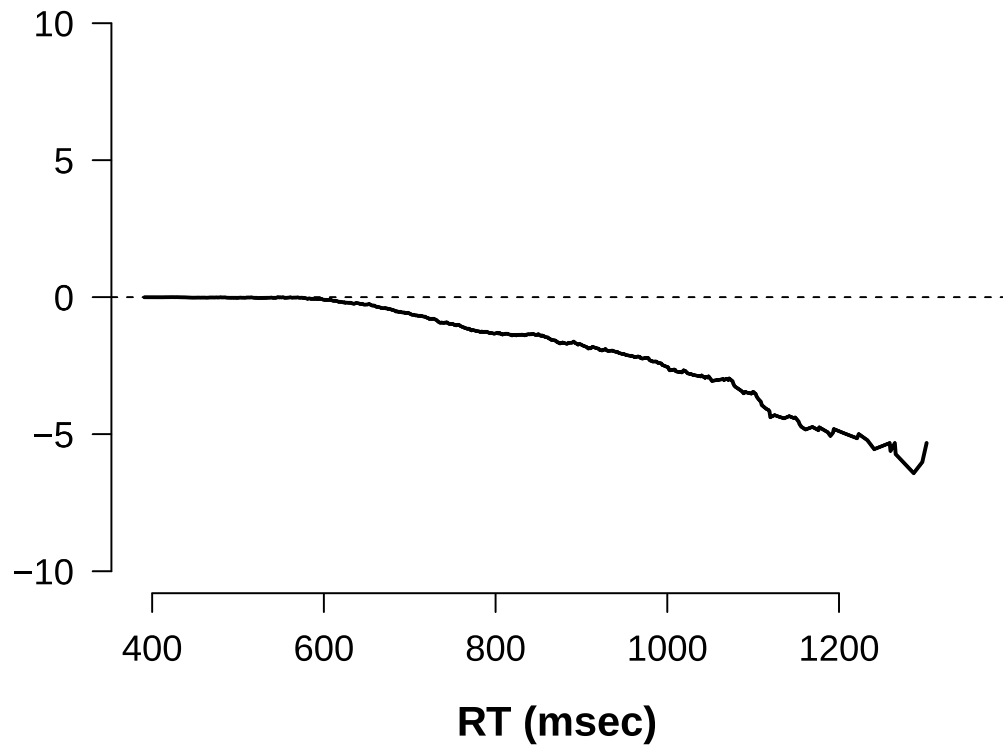
Subject 4



Subject 5



Subject 6



Subject 7