Sebastian Musslick, Amitai Shenhav, Matthew Botvinick, Jonathan Cohen

The Expected Value of Control Model
GREEN
Expected Value of Control Theory

\[
EVC(signal, state) = \sum_i \Pr(outcome_i \mid signal, state) \cdot Value(outcome_i) - Cost(signal)
\]

Shenhav, Botvinick, & Cohen (2013)
Expected Value of Control Theory

GREEN

Payoff vs. Control Signal Intensity

- Increased payoff
- EVC

Cost
Expected Value of Control Theory
Expected Value of Control Theory

Neural Architecture

Shenhav, Botvinick, & Cohen (2013)
Parameterize control task environment in terms of accumulation-to-bound model (e.g., DDM)

Map control intensity to linear changes in one or more parameters of this model, with attendant influence on error rate and RT
Parameterize control task environment in terms of accumulation-to-bound model (e.g., DDM)

Map control intensity to linear changes in one or more parameters of this model, with attendant influence on error rate and RT

Map control intensity to some monotonic cost function

Set parameter(s) so as to maximize value (function of reward, error rate, and/or RT) and minimize costs based on (imperfect) information about environment

Learn about environment based on feedback
**EVC Model**

Implementation

- **set up EVC model**
- **build actual state**
- **prime expected state**
- **calculate EVCs**
- **apply signal w/ max. EVC**
- **update expected state**

**External Task Environment**

**Cognitive System** (Internal Model)
EVC Model

Implementation

set up EVC model

build actual state

prime expected state

calculate EVCs

apply signal w/ max. EVC

update expected state

Run *expected/internal* decision model (generate predicted outcomes)

Run *actual* decision model (implement decision in actual world)
Implementation

**EVC Model**

- Set up EVC model
- Build actual state
- Prime expected state
- Calculate EVCs
- Apply signal w/ max. EVC
- Update expected state

**Task environment**
- Trial structure
- Outcome values
- Stimulus-response mappings

**Individual system parameters**, e.g.
- Control signals
- Cost functions
- EVC discount factor
- Learning rate
EVC Model

Implementation

- set up EVC model
- build actual state
- prime expected state
- calculate EVCs
- apply signal w/ max. EVC
- update expected state

E.g. Erikson Flanker task

- stimuli
  - >
  - <
  - >
  - stim. salience
  - [0.3, 0.4, 0.3]

- stimulus-response mapping

- responses
  - Left
  - Right
  - outcome values
  - [3.0, -2.0]
EVC Model

Implementation

- Set up EVC model
- Build actual state
- Prime expected state
- Calculate EVCs
- Apply signal w/ max. EVC
- Update expected state

Expected state - Internal model of task environment
- Task difficulty (stimulus salience)
- Stimulus-response mapping
- Outcome values
EVC Model

Implementation

1. Set up EVC model
2. Build actual state
3. Prime expected state
4. Calculate EVCs
5. Apply signal w/ max. EVC
6. Update expected state

\[ EVC(signal, state) = -\sum_i \Pr(outcome_i | signal, state) \cdot Value(outcome_i) - Cost(signal) \]

DDM implementation

- Threshold for controlled response
- Drift rate
- Noise
- Starting point
- \[ drift = \sum_i^n Ctrl0_i + SignalToCtrl_i \cdot Intensity_i \]
- \( n \)... number of active signals
- Threshold for automatic response
EVC Model

Implementation

set up EVC model

build actual state

prime expected state

calculate EVCs

apply signal w/ max. EVC

update expected state

$EVC(signal, state) = \sum_i \Pr(outcome_i | signal, state) \cdot Value(outcome_i)$

accumulated costs for each control signal

control cost for each signal

signal intensity

$\sum \text{cost}$
**EVC Model**

**Implementation**

- **set up EVC model**
- **build actual state**
- **prime expected state**
- **calculate EVCs**
- **apply signal w/ max. EVC**
- **update expected state**

**Specification**

\[ \text{signal}^* = \max_i [EVC(signal_i, state)] \]

**Task execution**

- Apply control signal combination with maximal EVC for current expected state
- Retrieve outcomes of actual state
**EVC Model**

Implementation

- **set up EVC model**
- **build actual state**
- **prime expected state**
- **calculate EVCs**
- **apply signal w/ max. EVC**
- **update expected state**

- Compare expected vs. actual outcomes
- Update internal state representation

**Diagram:**
- Threshold for controlled response
- Drift rate
- Noise
- Shift starting point (= adjust expected trial difficulty)
- Threshold for automatic response
EVC Model

Implementation

EVC parameters

- control parameters
  - signal A intensity
  - signal B intensity
  - signal C intensity

Mapping function

- drift rate
  - $f_{Ctrl1}(x, \tilde{p}_{Model})$
  - $f_{Ctrl2}(x, \tilde{p}_{Model})$
  - $f_{Ctrl3}(x, \tilde{p}_{Model})$

DDM parameters

- threshold
- bias

Trial Simulation

state parameters

- target saliency
  - $f_{State}(x, \tilde{p}_{Model})$
EVC Model

Example Simulation

- Actual state param (red) and expected state param (green)
- Control intensity
- Trials
- Difficulty
- Ctrl Intensity
- Trials
Simulations & Predictions

Manipulating Reward

**Expected Value of Control**

- **EVC**
- **Trials**

**Reward Manipulation**

- **Reward**
- **Trials**

**Control Signal Adjustments**

- **Intensity**
- **Trials**
Simulations & Predictions

Manipulating Reward

- **Intensity Adjustment (trial n+1)**
  - Δ Intensity
  - Increments (Incr): 0.15
  - Decrements (Decr): 0.1

- **RT Adjustment (trial n+1)**
  - Δ RT (ms)
  - Increments (Incr): 0.15
  - Decrements (Decr): 0.1

- **ER Adjustment (trial n+1)**
  - Δ ER (%)
  - Increments (Incr): 0.02
  - Decrements (Decr): 0.01
Simulations & Predictions

Manipulating Trial Difficulty

![Graphs showing error rate, stimulus saliency manipulation, and control signal adjustments over trials.](image)
Simulations & Predictions

Manipulating Trial Difficulty

Control Signal Adjustments

Cost Function
Simulations & Predictions
Sequential Control Adjustment
Simulations & Predictions

Task Switching

attractor cost function

configuration costs

euklidian distance

costs

signal 2

(0.3; 0.3)

(0.8; 0.5)

signal 1

0 1 2 3 4 5

0 0.1 0.2 0.4 0.6 0.8 1.0
Simulations & Predictions

Task Switching

**Trials**

<table>
<thead>
<tr>
<th>Difficulty</th>
<th>Trials</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1-2</td>
</tr>
<tr>
<td>0.5</td>
<td>3-4</td>
</tr>
<tr>
<td>1</td>
<td>5-8</td>
</tr>
</tbody>
</table>

**Control Intensity**

- Signal "parity": orange
- Signal "magnitude": blue

**Parity Task**

- **Parity Signal**: even
- **Response**: 1 (correct) if even, 2 (incorrect) if > 5

**Magnitude Task**

- **Signal**: even
- **Response**: 1 (correct) if even < 5, 2 (incorrect) if even > 5

**Example**

- **Trial 1**: Parity Signal: even, Response: 1 (correct)
- **Trial 2**: Magnitude Signal: even, Response: 2 (incorrect)
Simulations & Predictions

Cued Task Switching
Simulations & Predictions

Cued Task Switching
Simulations & Predictions

Configuration Costs & Sequential Adjustments
Simulations & Predictions

Cued Task Switching

Umemoto & Holroyd (2014)

EVC model
Simulations & Predictions

Voluntary Task Switching

- **Reward Reversal**
  - Graph showing control allocation over trials with different reward phases.

- **Adaptation vs. Avoidance**
  - Graph comparing total control allocation over trials with a specific trial marked as '6 = 0'.

The graphs illustrate the patterns and changes in control allocation under voluntary task switching scenarios.
 Experimental Investigations

Task Switching

cued task switching block

add

\[ \frac{36}{52} \checkmark \]

subtract

\[ \frac{11}{14} \checkmark \]

individual model fit

EVC Model
Experimental Investigations

Task Switching

voluntary task switching block

control
choice behavior

predict
choice behavior & performance

EVC Model

?  x  \frac{46}{54}  \checkmark

?  x  \frac{33}{18}  \checkmark