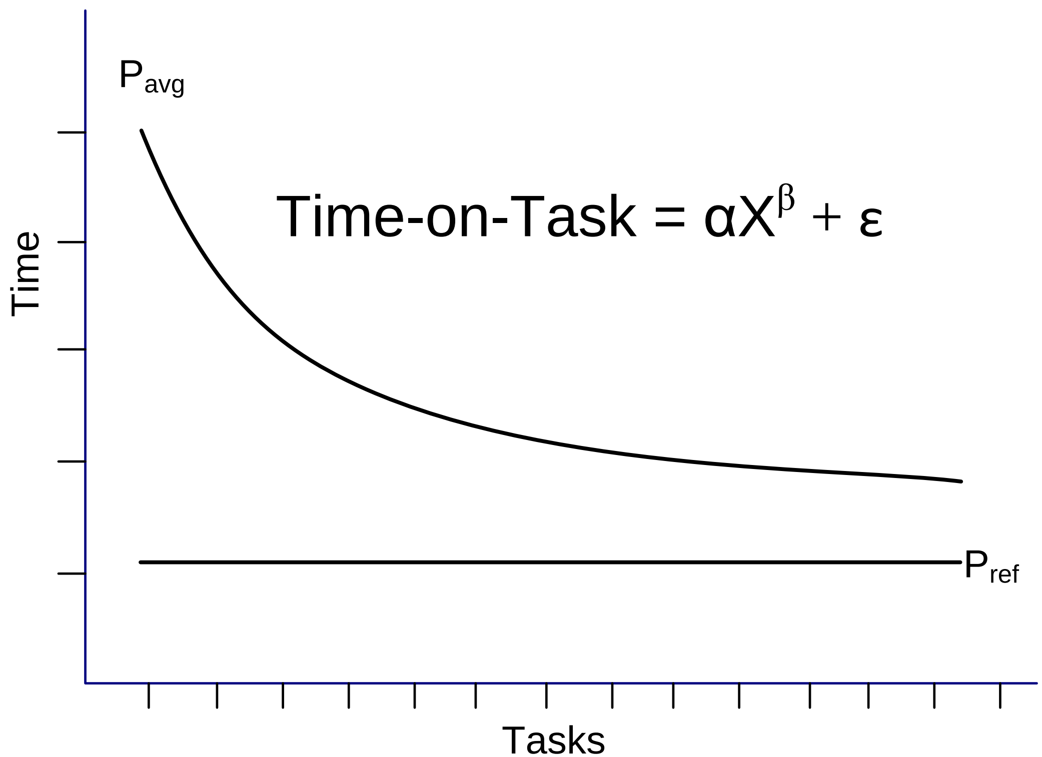
Measuring Usability

Mueller, Komogortsev, & Tamir

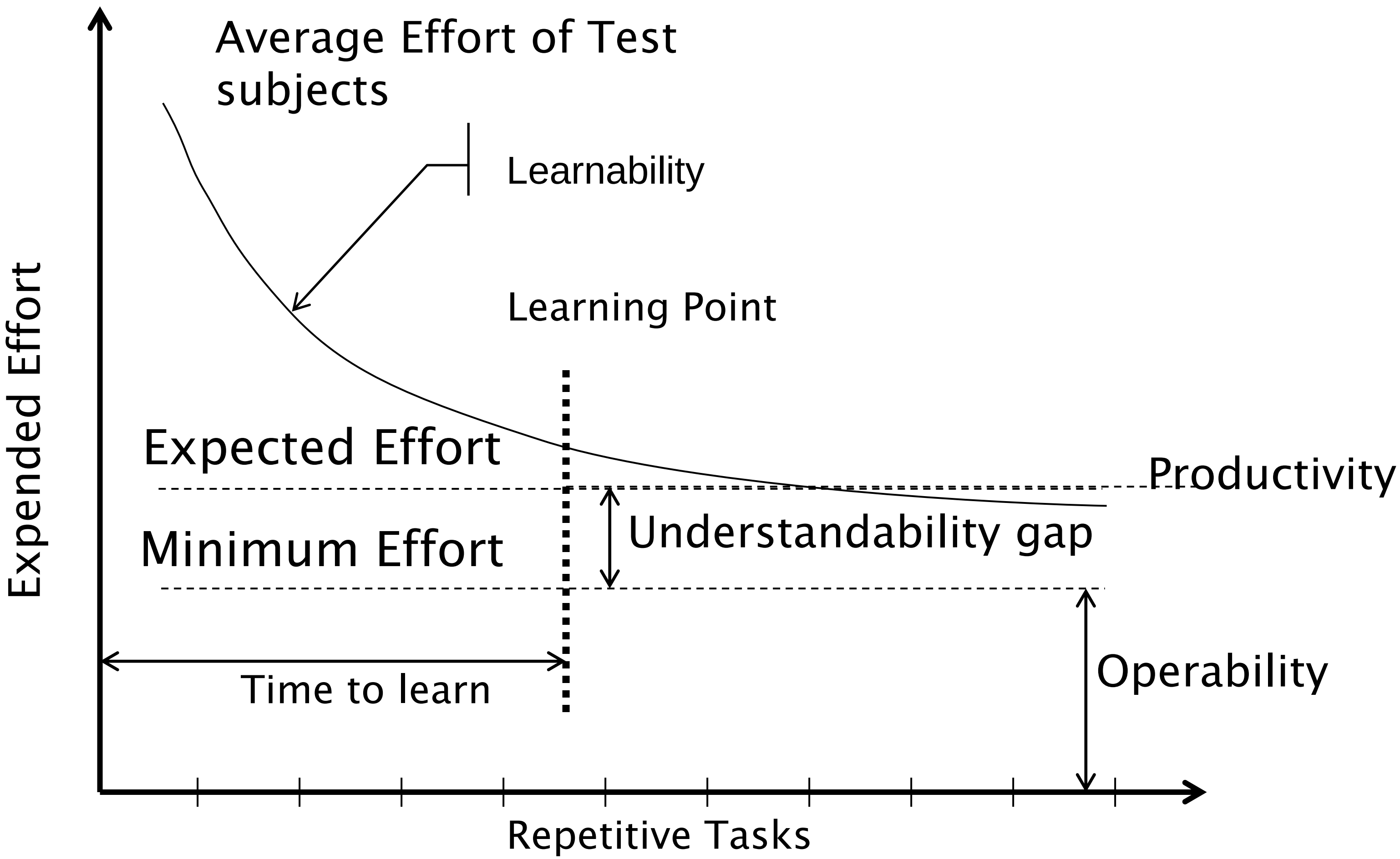
Observations

- Usability is inversely proportional to effort
  - User effort is related to manual effort – e.g., number of mouse clicks, number of key-board clicks, mouse path traversed.
- A set of identical independent ("iid") experiments on a single scenario can be used to measure learnability and operability
- Eye tracking can be used to provide additional measures of physical and manual effort

Traditional Learning Curve

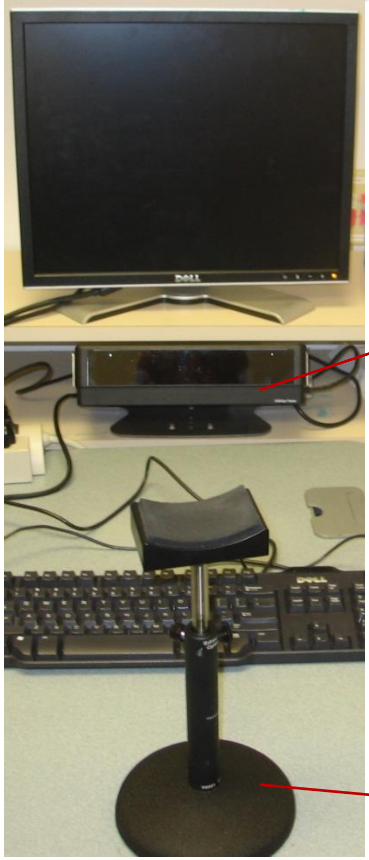


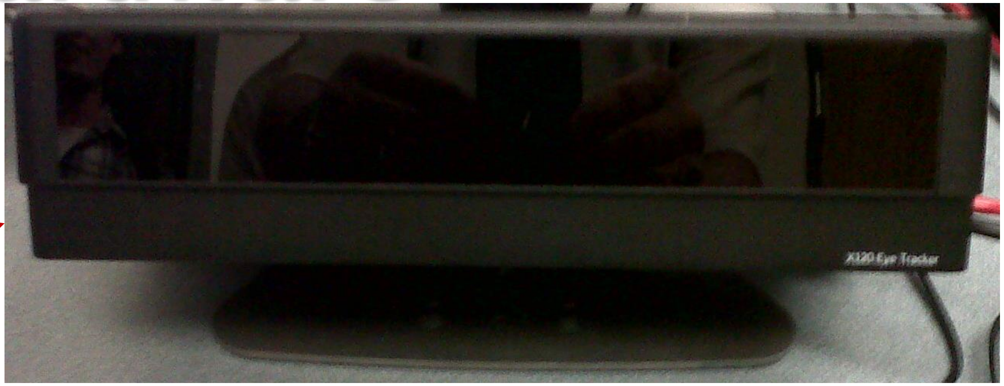
Effort-based Usability Model



*Based on ISO/IEC 9126-1:2001 Standard

Eye Tracker Hardware



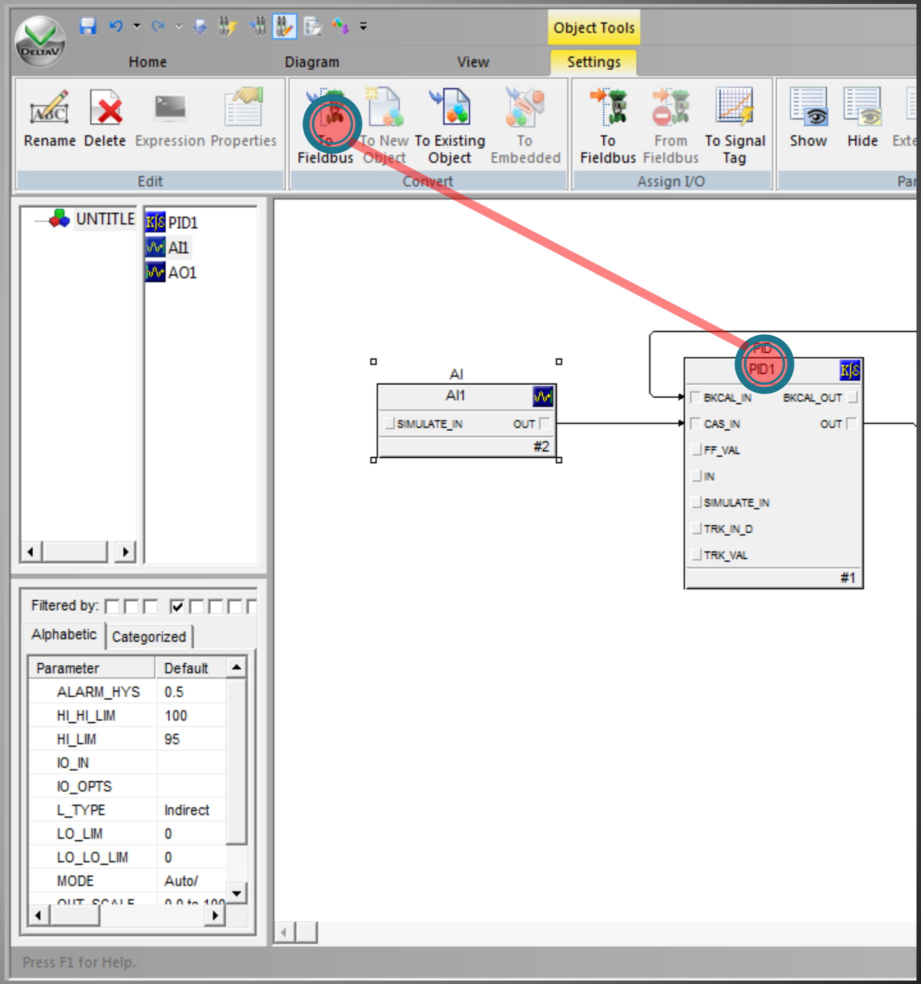


Eye tracker

Chin rest

Fixations and Saccades

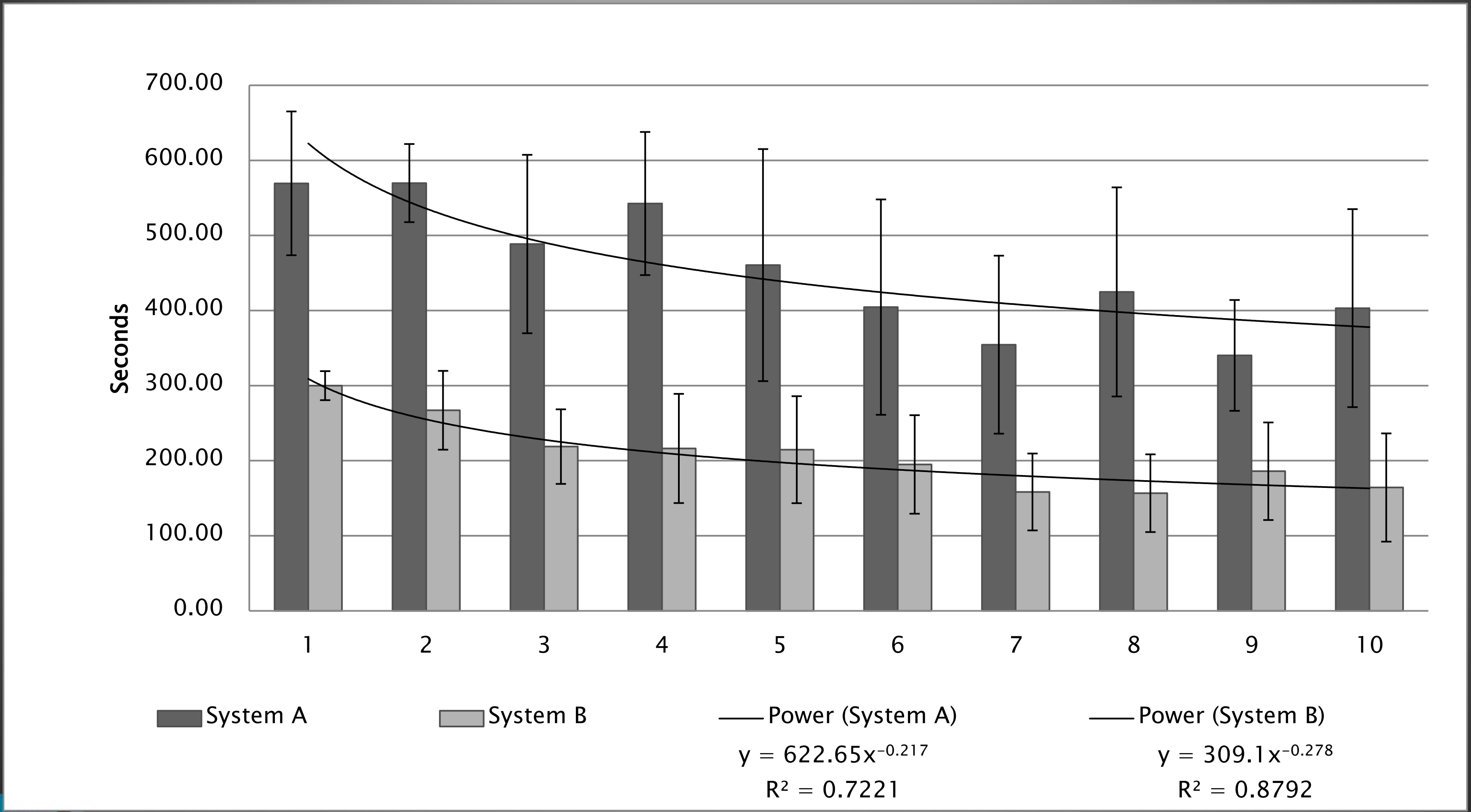
- When performing a task, fixations and saccades can reflect effort expended.
- Greater effort =
  - Longer fixation duration
  - More fixations
  - Longer saccade length
  - More saccades



Measurements

- Time on Task
- Number of Mouse/Keyboard clicks
- Total mouse path traversed
- Average fixation duration
- Average pupil diameter
- Number of fixations
- Average saccade amplitude
- Number of saccades
- Total eye path traversed

Travel Reservation Experiment Time on Task



Pilot Project

Emerson / TxState Usability Experiment

- Purpose
  - Pilot Study to determine the usefulness of the Texas State University methodology in measuring aspects of Usability in Emerson products

- Primary Goal
  - Compare the usability of a limited set of tasks in two versions of Control Studio referred to as **System A** and **System B**

Scenario-based Test Design

- Our test consisted of 15 repetitive tasks.
- Each task followed the same general workflow, but the function blocks, parameters, and properties being worked on were varied. IID Tasks
- The task instructions were written in general terms such as “Add an AI block” but did not specify how to carry out the work.

Scenario-based tasks used in Experiment

Appendix C Tasks

TASK 1

<Start>

1. Delete block PT3-15 from the Distillation Column COLUMN1.
2. Add an Analog Output to the right of the block PIC3-15 and name it as VENT_VALVE.
3. Make the following connections -
   a. VENT_VALVE **OUT** to PIC3-15 **BKCAL_IN** and set the connection as feedback
   b. PIC3-15 **OUT** to VENT_VALVE **CAS_IN**
4. Transfer the changes to the Controller Simulator. Change Control Studio to view the information from the Controller Simulator
5. Change the PIC3-15 Pressure control set point (SP) to 25.
6. Change Control Studio to view the information in the Configuration Database
7. Upload and save the changes

<End>

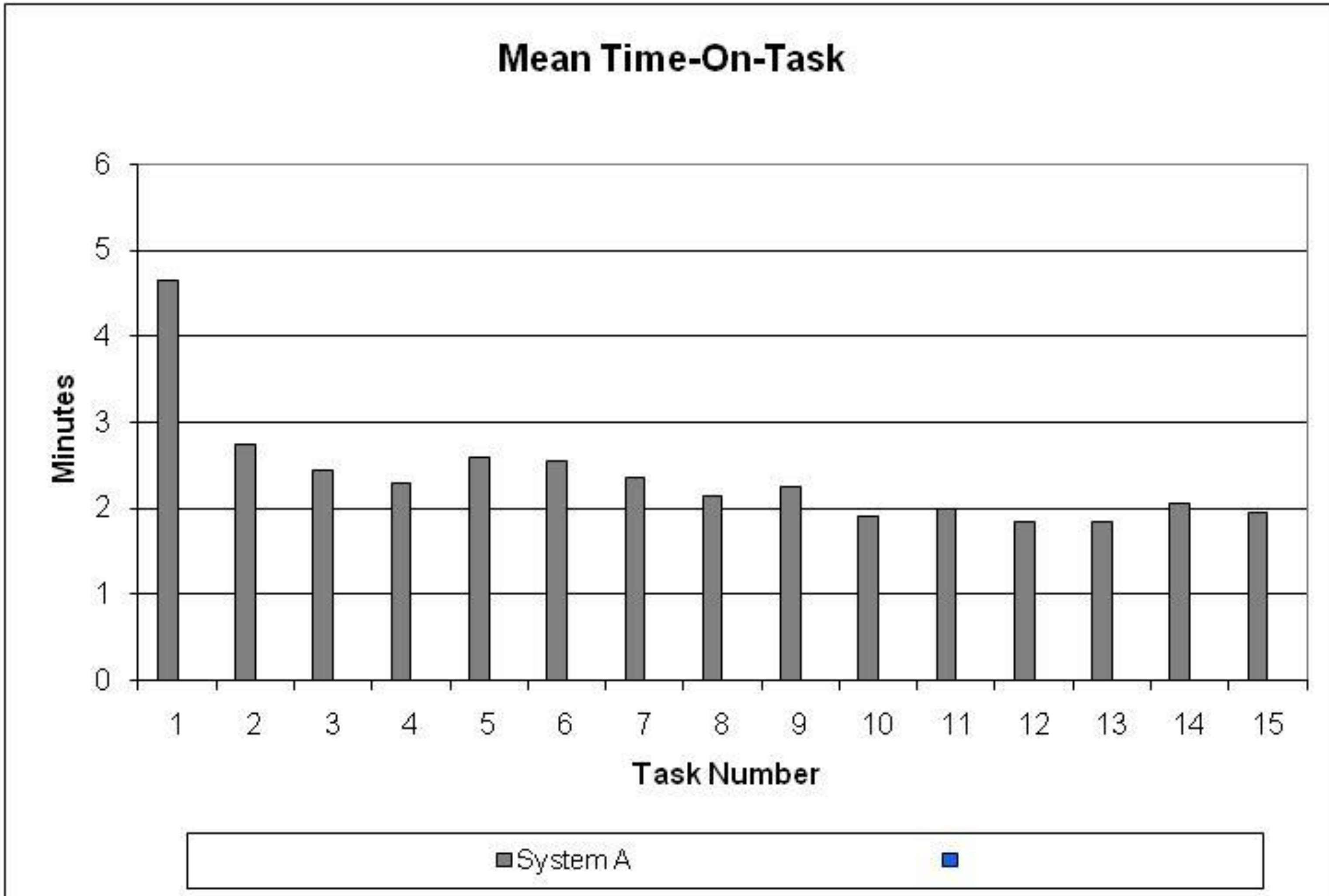
TASK 2

<Start>

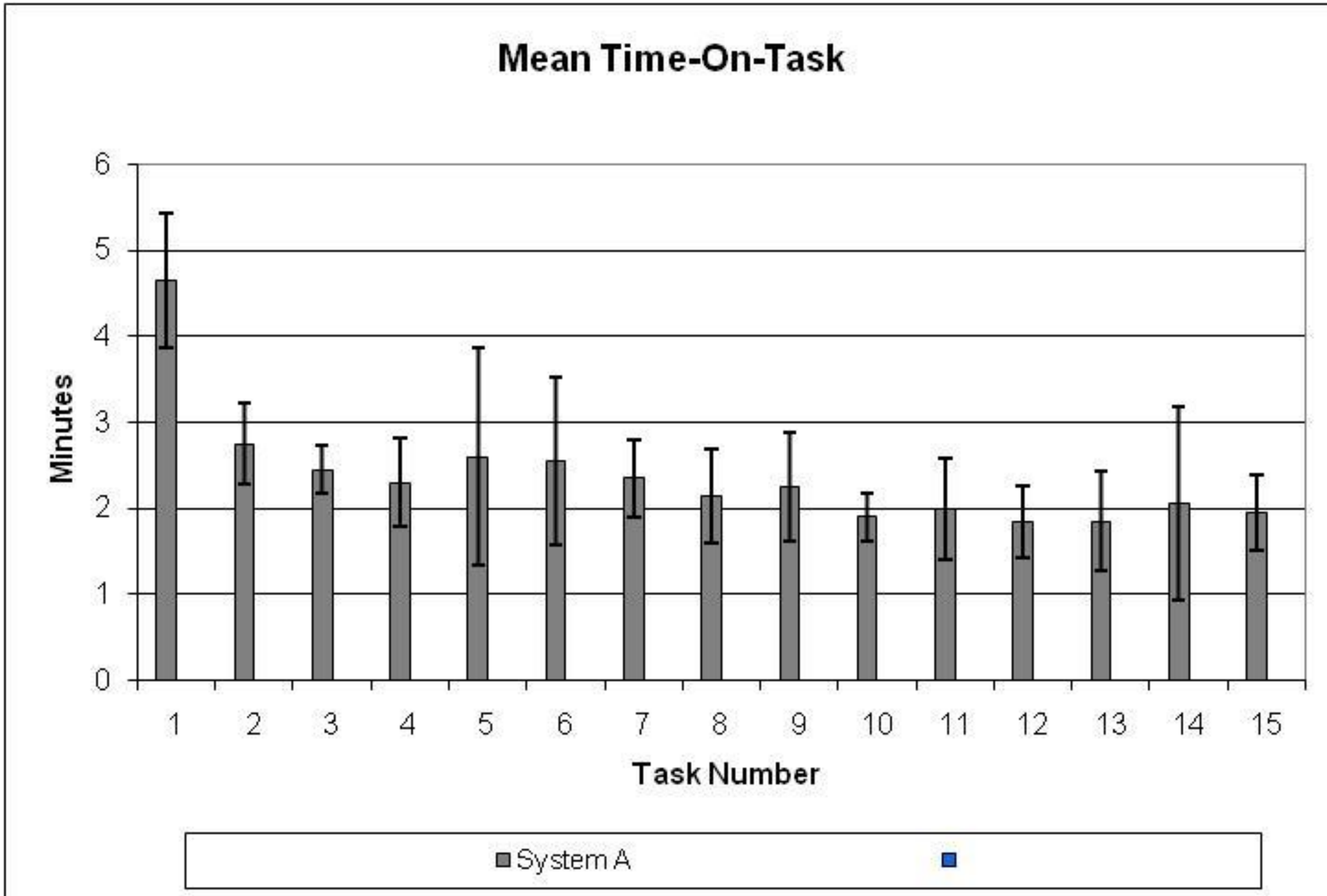
1. Delete block LIC3-16_RSP from the Distillation Column COLUMN1.
2. Add an Analog Input to the left of the block PIC3-15 and name it as PT3-15.
3. Make the following connections -
   a. PT3-15 **OUT** to PIC3-15 **IN** and set the connection as feedback
4. Transfer the changes to the Controller Simulator. Change Control Studio to view the information from the Controller Simulator
5. Change the VENT_VALVE SP_HI_LIM to 85
6. Change Control Studio to view the information in the Configuration Database
7. Upload and save the changes

<End>

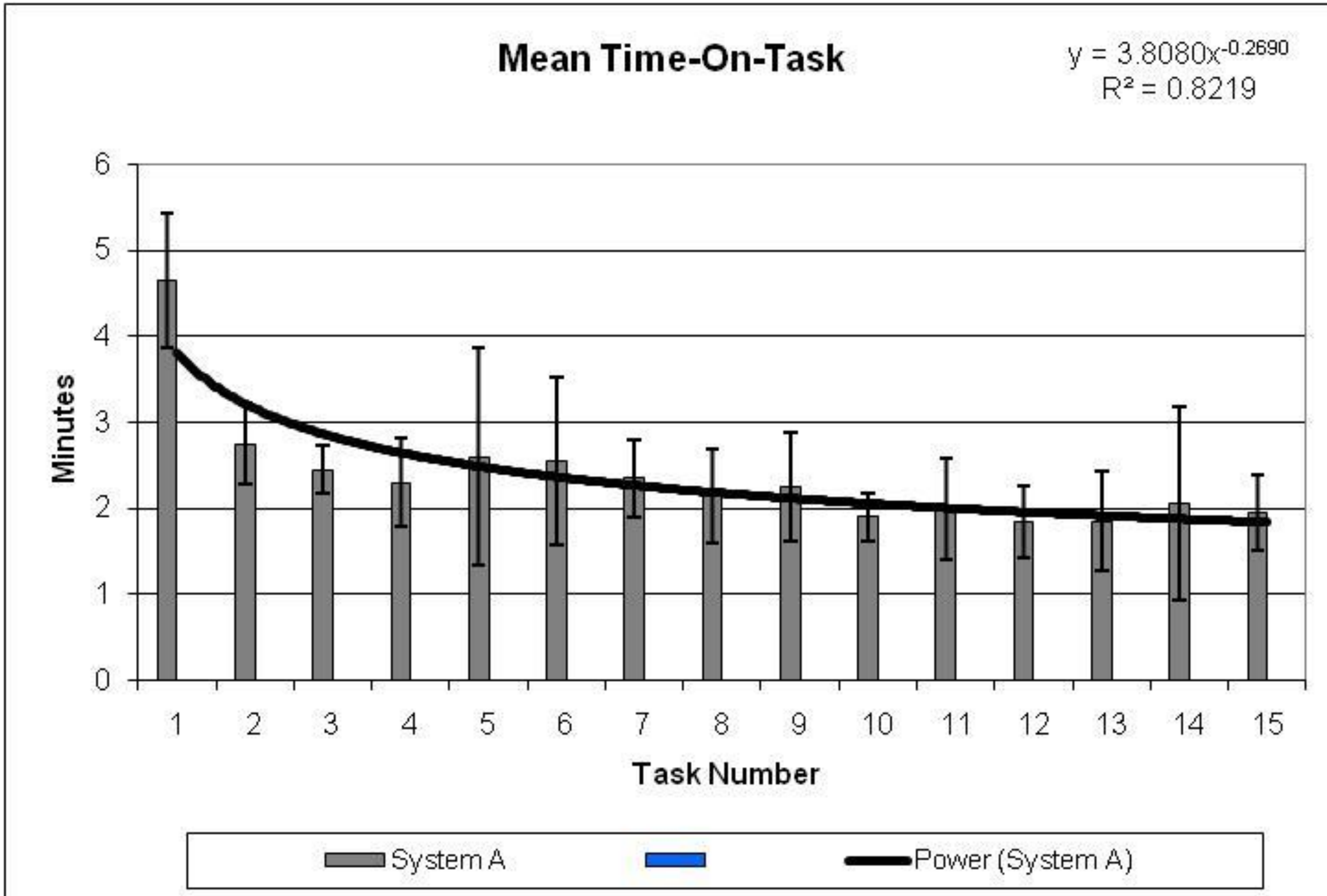
Mean Time to Complete a Task in System A



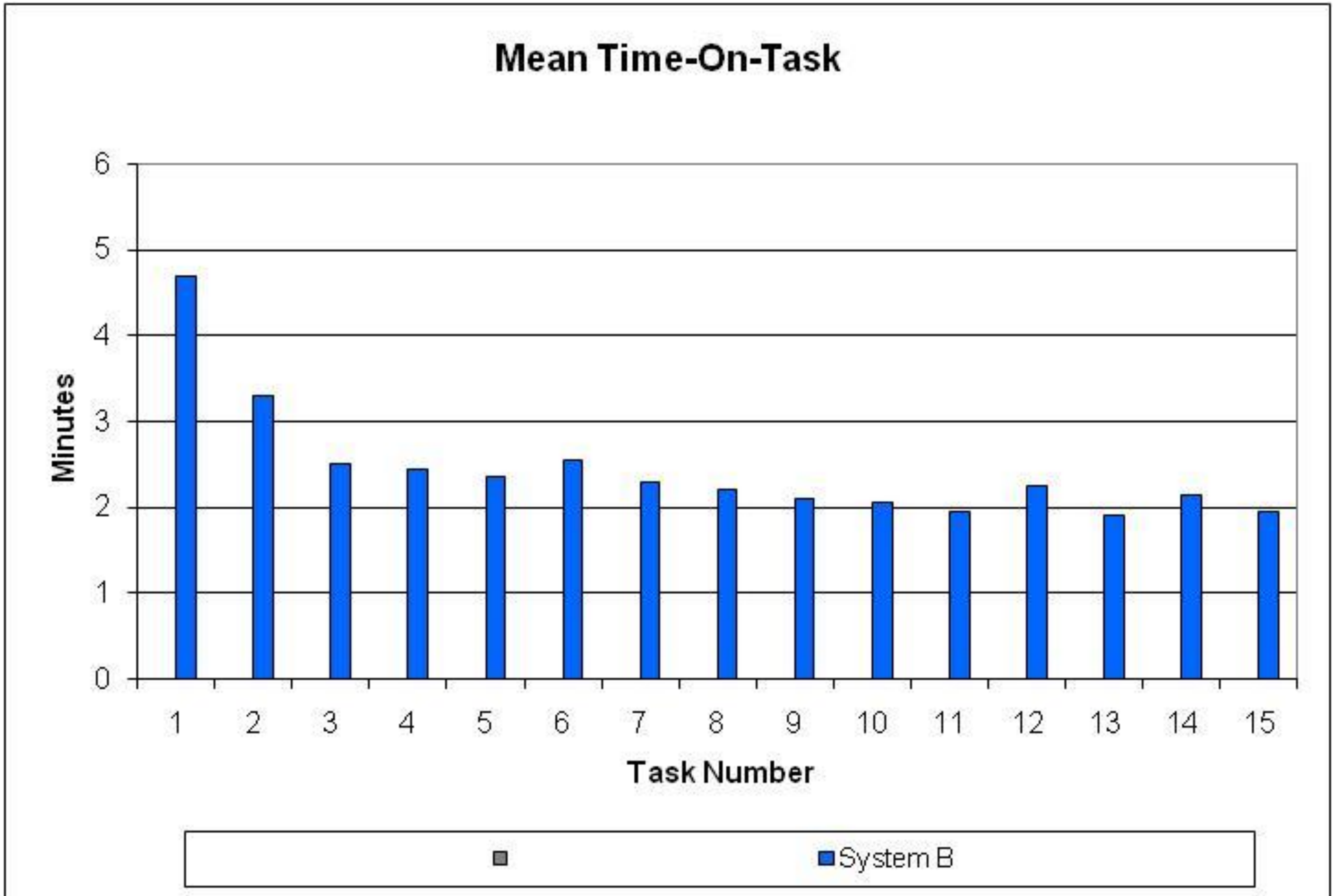
Standard Deviation for a Task in System A



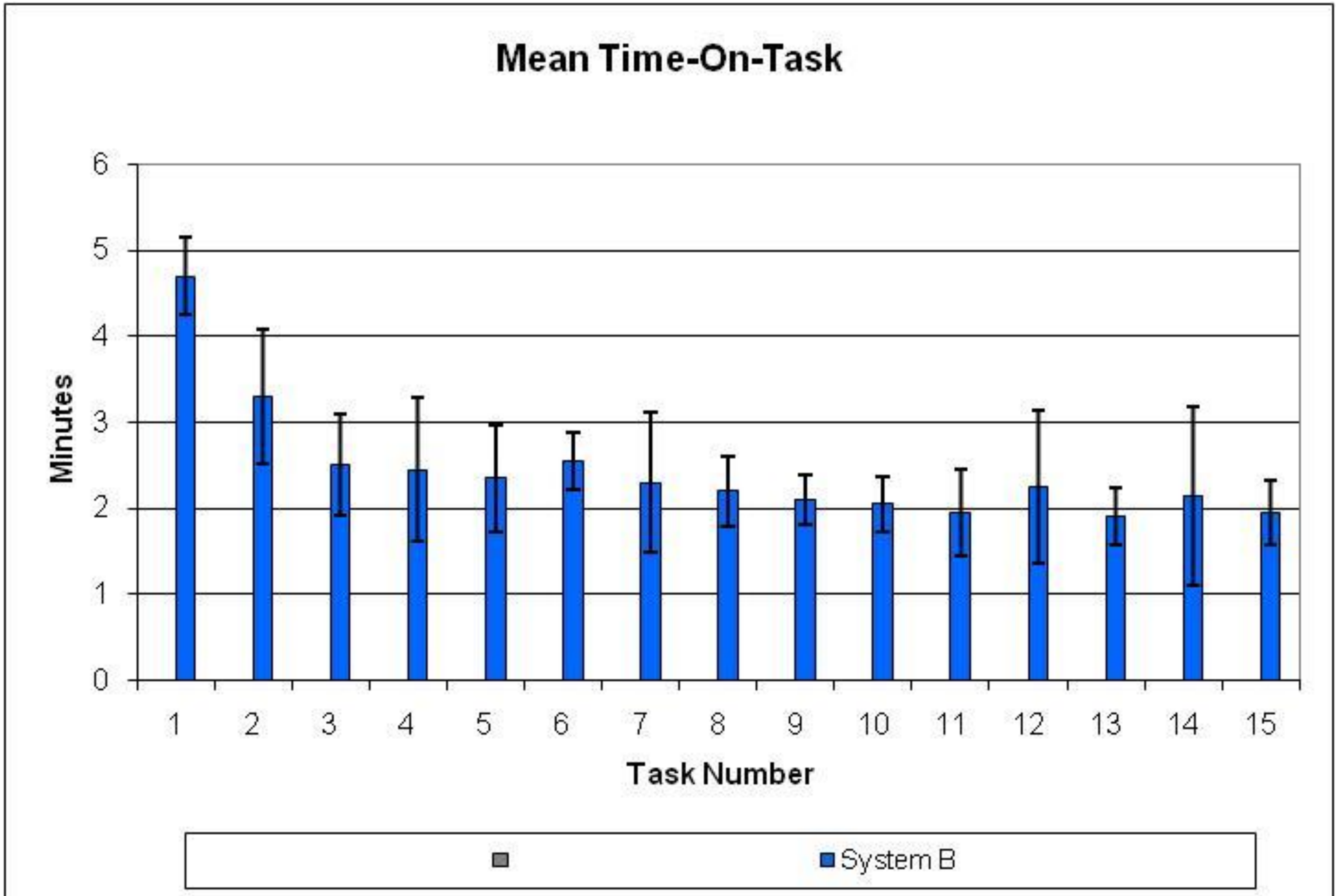
Power Curve Matching Tasks of System A



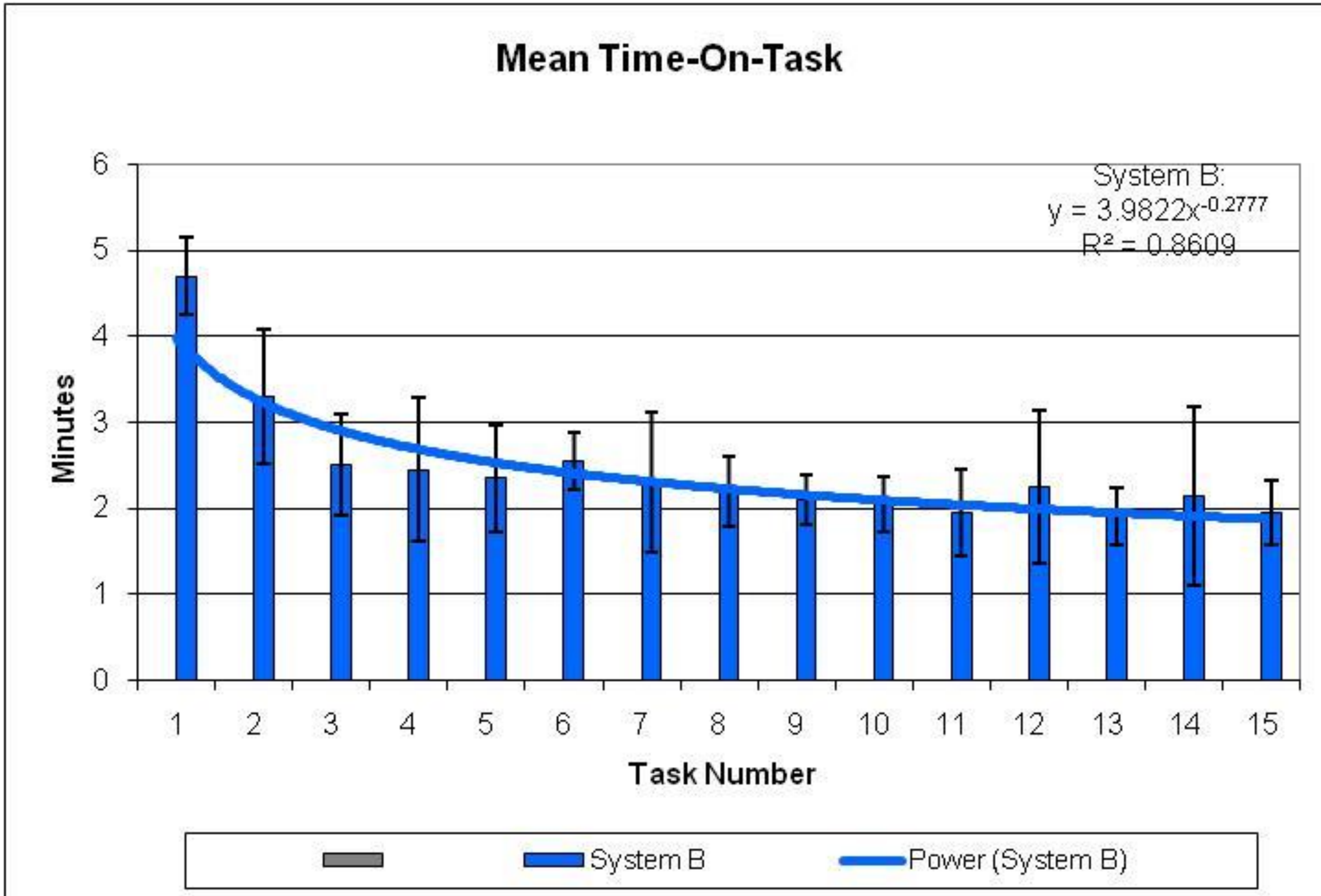
Mean Time to Complete a Task in System B



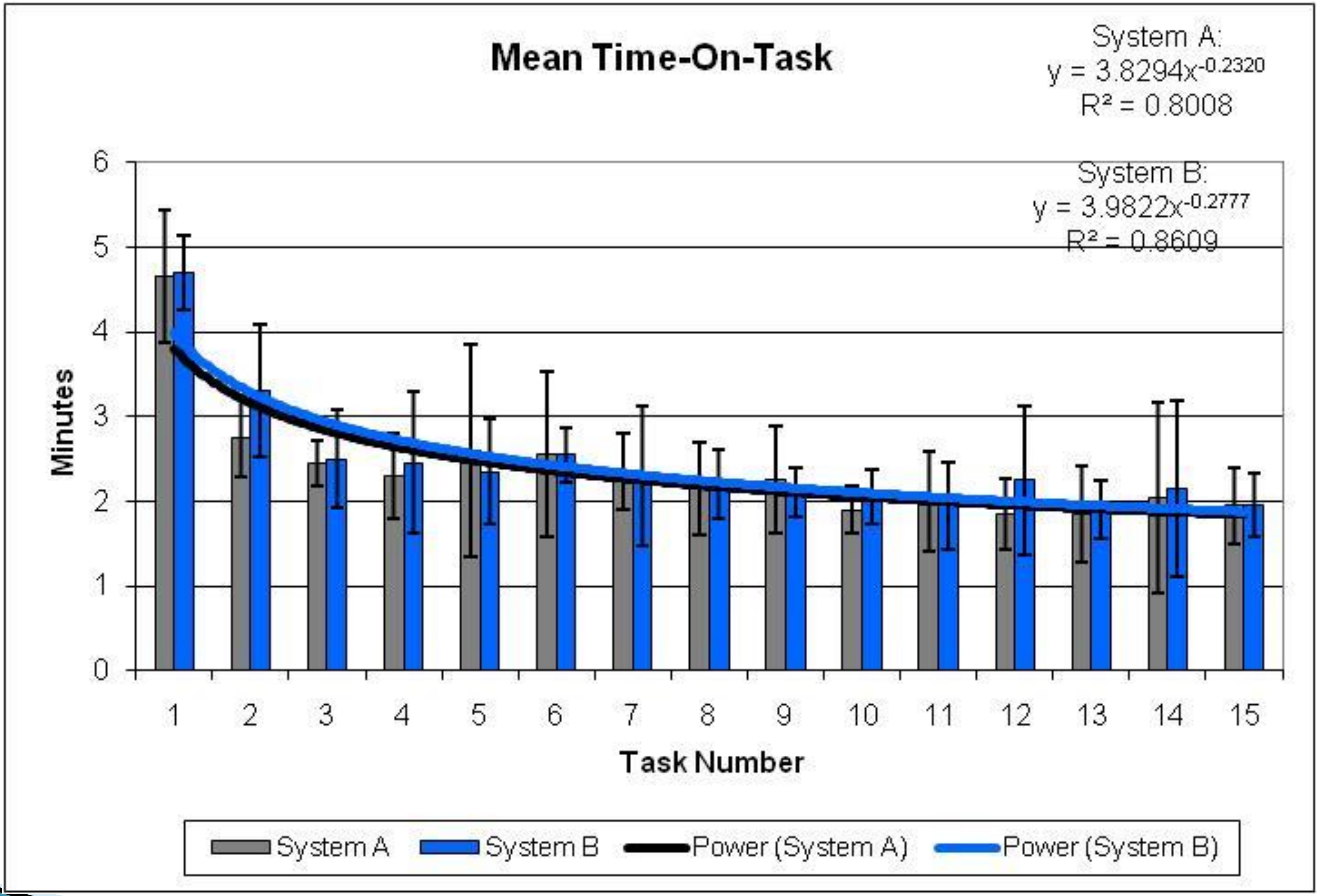
Standard Deviation for a Task in System B



Power Curve Matching Tasks of System B

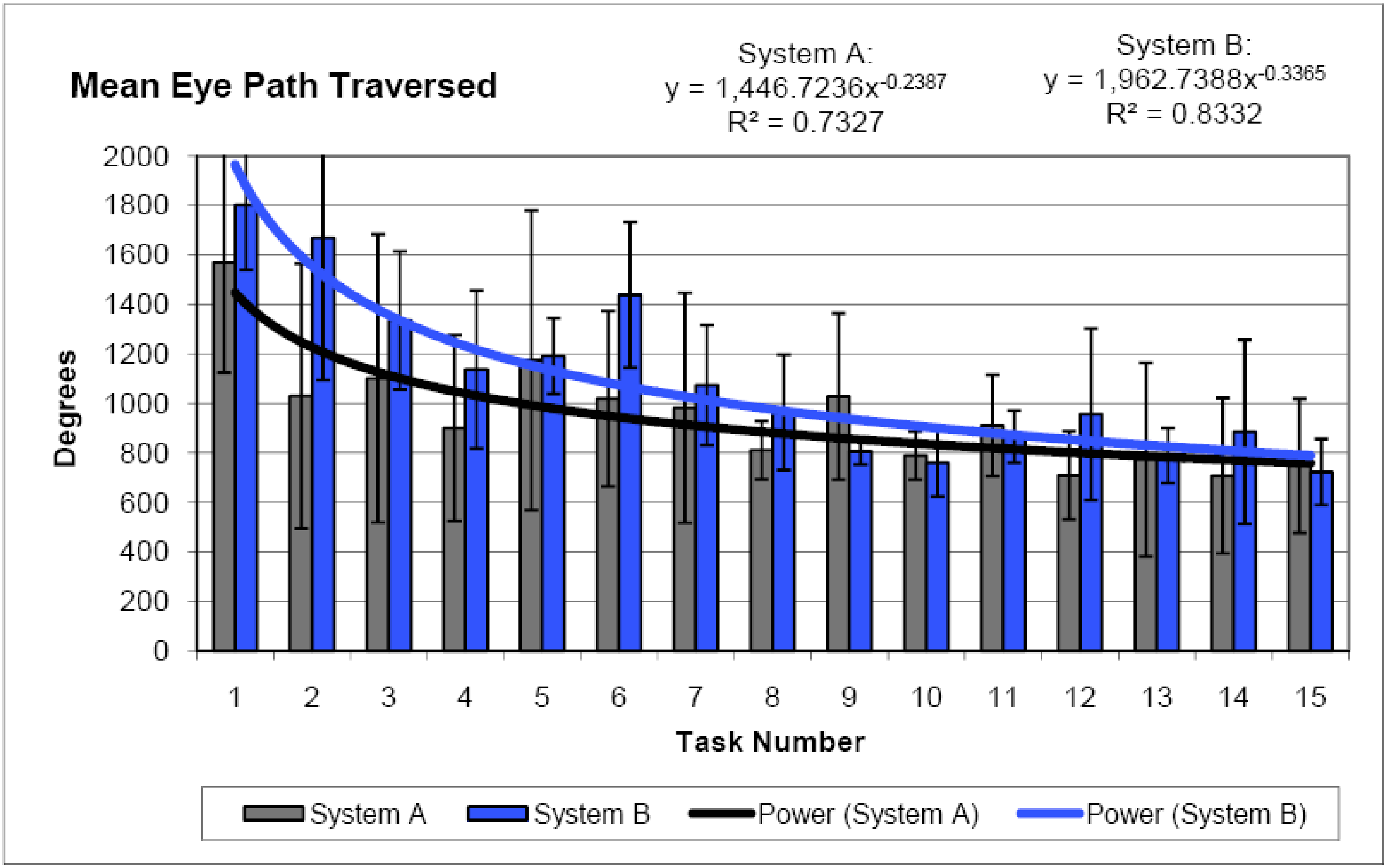


Overall Learnability



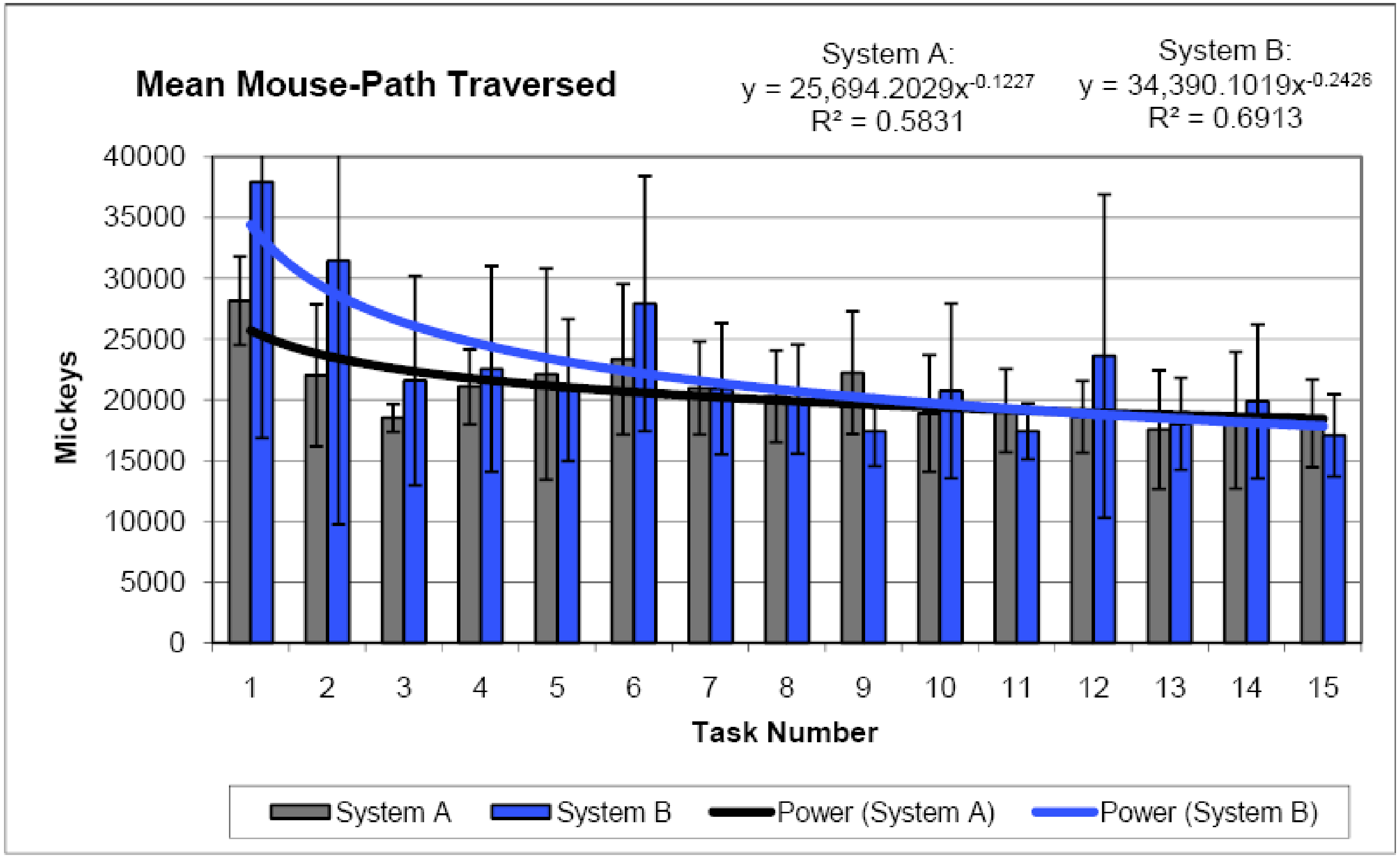
Content contributed by Texas State University

Physical Effort



Content contributed by Texas State University

Physical Effort



Content contributed by Texas State University

Experiment Conclusions

- A methodology involving eye tracking is a viable tool for objectively measuring usability
- After Learning point is reached, both System A and B have very similar usability characteristics
- People are able to learn to use the application with the updated user interface
- [After moderate training] student performance is close to “real user’s” performance

Current / Next Phases

- Phase 2
  - Analysis of additional scenarios using current Emerson software and prototypes of “next generation software”.
- Phase 3
  - Pinpoint analysis

Pinpoint Analysis

Current / Next Phases

- Phase 2
  - Analysis of additional scenarios using current Emerson software and prototypes of “next generation software”.
- Phase 3
  - Pinpoint analysis

Pinpoint analysis

$$R = \begin{bmatrix} r_1 \\ r_2 \\ . \\ . \\ r_j \\ \end{bmatrix}$$

$r_1 =$ Average Saccade Amplitude

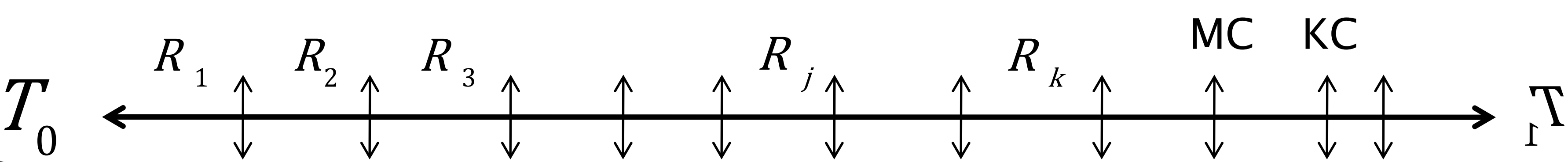
$r_2 =$ Average Fixation duration

$r_j =$ Total Time on Task

$$T_0 \longleftrightarrow T_1$$

Pinpoint analysis

- Semi-supervised
- Segment the data
- Use pattern recognition techniques to identify excessive- effort segments
  - Thresholding
  - Clustering (K-means)
    - Exhaustive feature selection
    - Principle component analysis
- Video clips corresponding to identified excessive-effort segments are further analyzed to spot usability issues



Pinpoint Analysis Phases

The pinpoint analysis includes two phases

- Training
- Classification

Training

- Set scenario
- Define iid tasks
- Determine the segmentation method uniform time or event driven
- Supervised feature (effort metrics) Selection (optional)
  - E.g., number of saccades per segment
  - An expert decides what metrics to use (e.g., ToT, mouse click)
- Select participants for generating the training data
- Produce training set (run tests and extract features for each segment)
- Unsupervised feature selection (optional)
  - E.g., use the add-up technique
- Set decision function per segment
  - E.g., a threshold on the average number of saccades per segment

Classification

- Select participants for generating the classification data
- For each participant
  - For each segment
    - Extract the selected feature (e.g., Tot in an event driven segmentation)
    - Measure the feature value the average (used as the threshold for the decision function in training)
    - Compare to threshold (in the pinpoint analysis group)
    - Make an Excessive (E) vs. Non-Excessive (NE) decision
- Get an expert to check segments classified as excessive effort segments

Supervised feature Selection

- An expert decided which feature[s] to use

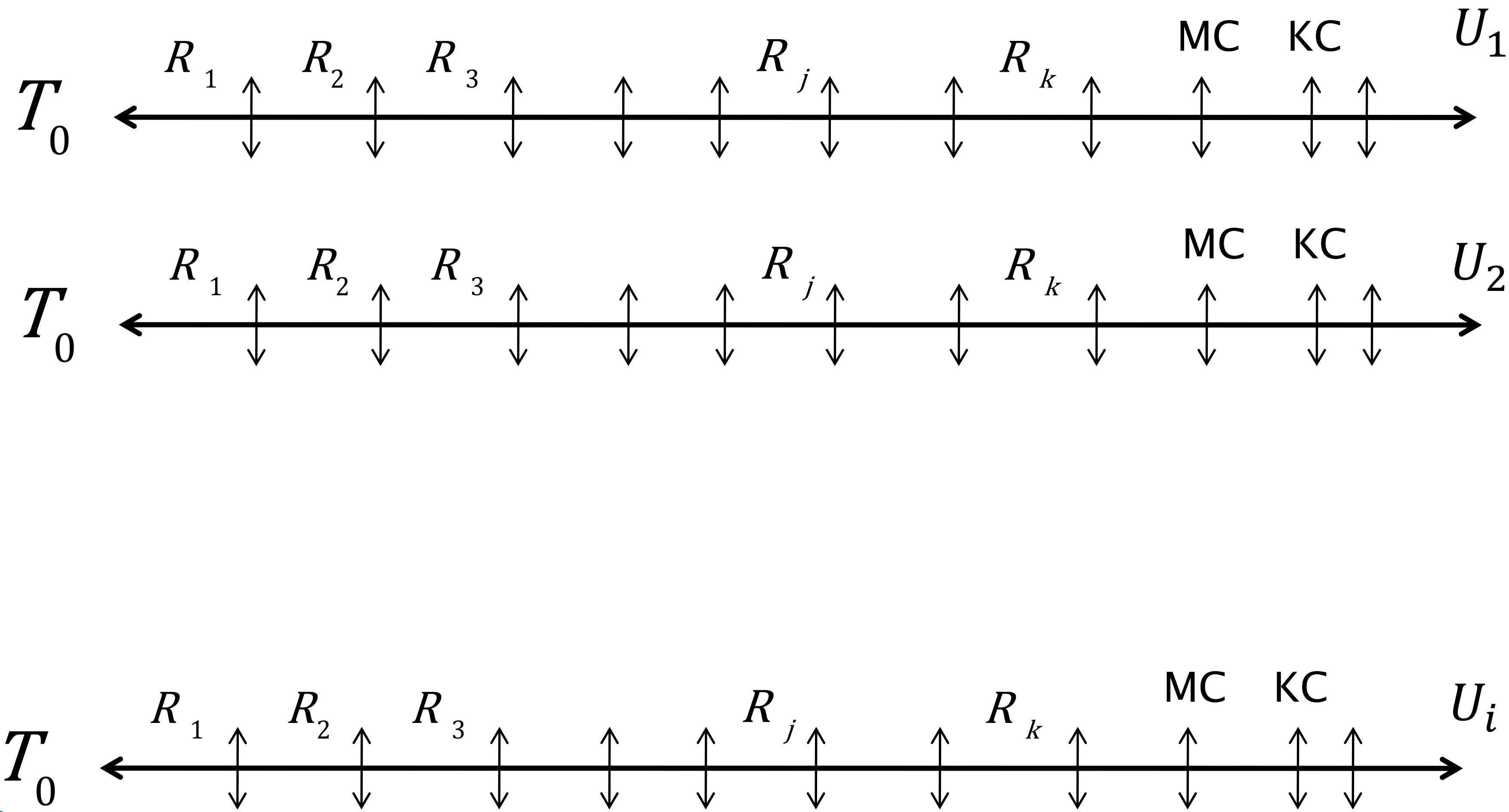
Unsupervised feature Selection

- Consider the training data
- For each participant
  - For each segment
    - Use a nock-down or add up approach
      - Extract a subset of features
      - Measure the features value
        - Cluster the data
        - Check the within (W) and between (B) dispersion metrics
- Choose the subset of features that produce best value for a given f(W)/f(B)

The nock down method

- Consider single features
  - Per segment cluster
    - Check the within and between dispersion metrics
    - Choose the singleton features that produce best value for f(W)/f(B)
- For each pair that contains the singleton chosen above
  - Per segment cluster
    - Check the within and between dispersion metrics
    - Choose the pair features that produce best value for f(W)/f(B)
- For each triple that contains the singleton chosen above
  - Per segment cluster
    - Check the within and between dispersion metrics
    - Choose the triplet features that produce best value for f(W)/f(B)
- Continue until reaching an upper limit on set cardinality

Illustration



The Sentic Mouse

Physiological Emotion applications

- MIT Affective Computing Lab's Affective Tangibles Program
- Mouse behaviors – number of mouse clicks, duration of mouse clicks

GOMS Research

- Goals, Operators, Methods, and Selection Rules (GOMS).
- A family of models to predict the usability of software for a task.
- Natural GOMS Language (NGOMS)
  - Learning time and execution time are predicted based on a program-like representation.
  - Assumes methods are strictly sequential and hierarchical in form (?).