**EPIC Tutorial Exercise 3: Exploring Multi-task Executives**

**David Kieras, Univ. of Michigan**

## I. Single Tasks

### A. We will Use a Subset of Wickens Task Conditions

**Fitting data is not the goal so much as learning how to explore the possibilities of massive parallelism possible in EPIC.**
Makes it easy to explore a variety of concepts of executive control.

**Easy tracking only.**

**Five levels of stimulus separation: 3.2, 6.4, 12.8, 22.4, and 35.2 degrees.**

**Device defaults to only 10 trials for testing and debugging purposes.**
Control/Run Controls/Device parameter string:

- number of trials
- stimulus separation (-1 for all)
- tracking difficulty: Easy, Diff, Both

### B. "WStarterRules.prs" Model for Wickens Task

**Fully Sequential Specialized Dual-Task Executive**
Controls which tasks run by which Goal items are in memory.
Controls tasks using particular task specifics.

**Tracking Task Rules**
Move the eye to the tracking target.
Do a Ply movement to put the cursor on to the target.

**Choice Task Rules**
Move the eye to the choice stimulus.
Select and perform the choice response.

**Identify Objects Startup Rules.**
Tag the three permanent screen objects.

**You can alter the Executive process (see comments)**
Run Tracking Task Alone - see also WTrackingOnly.prs
Run Choice Task Alone - see also WChoiceOnly.prs
Run both sequentially - as is.

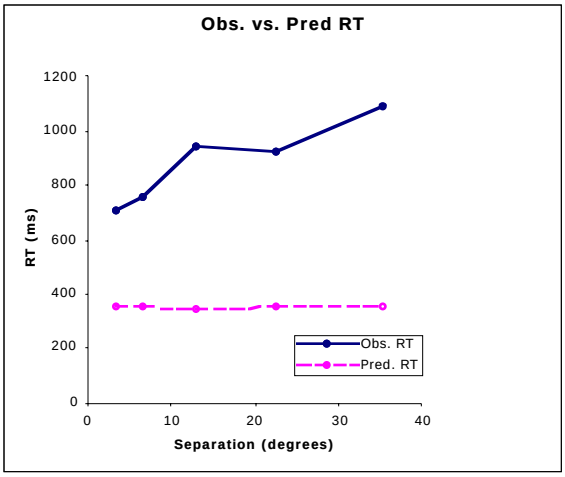
C. **Running the Single-Task Models**

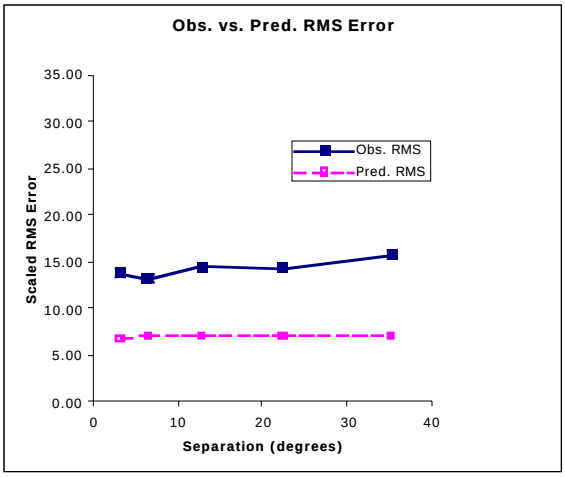
**Modify the executive process to run each task alone.**

See pre-modified rule sets, WChoiceOnly.prs, WTrackingOnly.prs
Plot the results and compare to observed dual-task results.

- Use Excel spreadsheet "WickensTemplate.xls"

**My results:**




Obviously no performance problems if only one task done at a time!

II. **Dual Tasks**

A. **Run the Fully Sequential Specialized Executive**

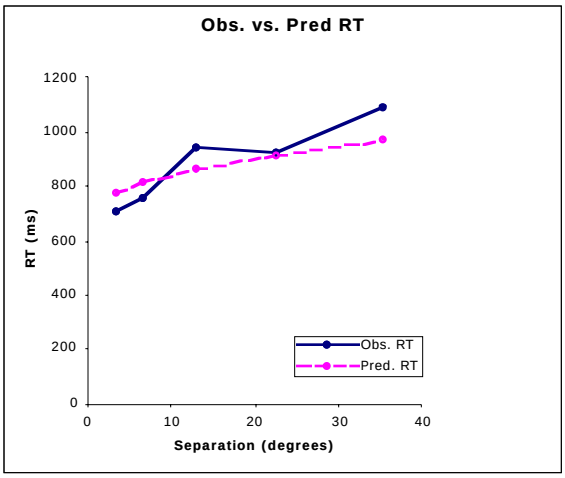
**Rules in "WStarterRules.prs"**

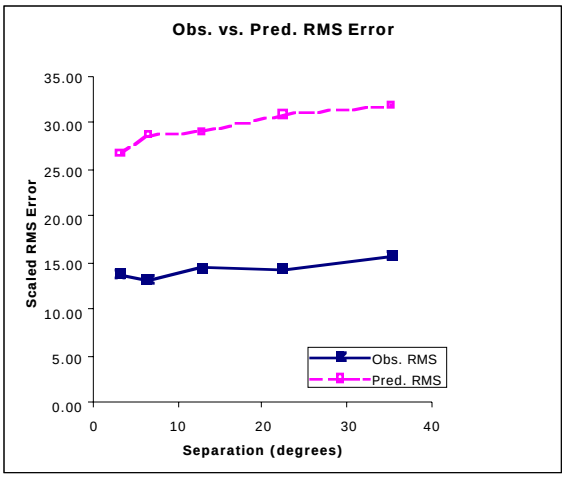
**Step through on single cycle to understand how it works.**

Use Controls/Run Controls break facility to save trouble.
Notice how each motor action must be complete before next step is done.

**Run for 100 trials and plot the results.**

**My results:**




Severe dual task deficit!
Where is it coming from?

B. **Possible Ways to get Better Performance**

**Basic problem: Tracking suspended during Choice Task execution.**
Tracking task suffers considerably from even a short delay.

**Speed up the choice task rules by overlapping motor processing:**
If choice task rules execute faster, tracking won't be as delayed.
See optional exercise.

**Optimized Specialized Executive:**
Executive could produce very fine-tuned interventions in the task rules.
- Modify task rules to produce more state information for executive.
- Modify task rules to await instructions from executive.

Disadvantage: Blurs distinction between task and executive rules.

**Single optimized rule set:**
Rewrite task rules so that all one big rule set, in which perceptual and motor parallelism is fully exploited.
No overhead-producing hierarchical method structure.
No unnecessary waiting for anything.
Disadvantage: Difficult to write; seems to imply considerable practice.

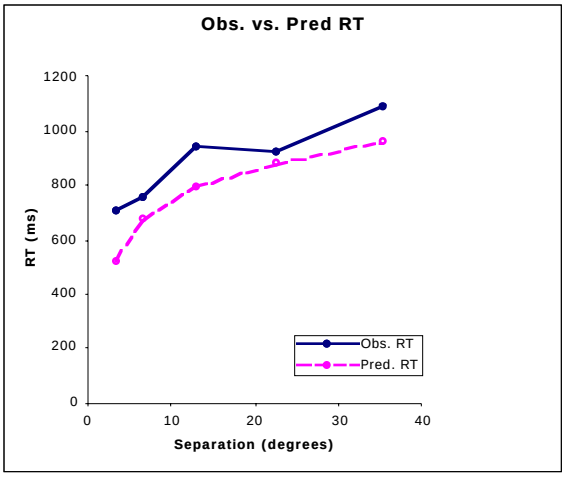
**Use Parallel General Executive to manage motor processors:**
Allocate processors to tasks on request, instead of suspending whole task.
Task rules must follow protocol, but otherwise remain simple and distinct.

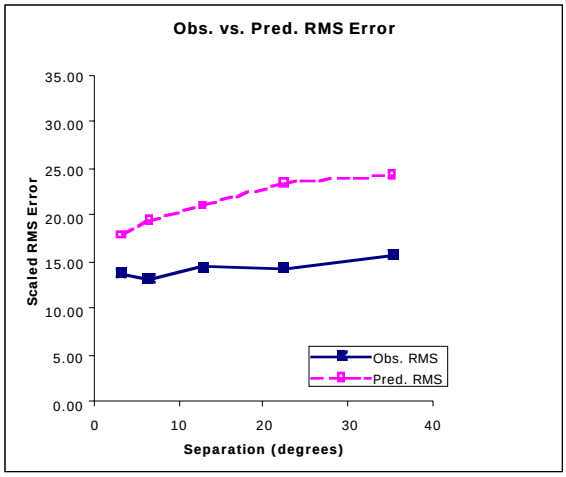
C. **Speeding up Sequential Model (Optional)**

**Motor processing can continue while cognitive processing goes on.**
Often no need for next step to wait for motor activity to be complete.
Can you eliminate some of the waits for Modality Free?

**Also, depending on separation, the Shape of the choice stimulus might be available prior to the eye movement to the choice stimulus.**
Can you modify the rules to take advantage of this?

**My results**




RT performance is now too fast, actually, and tracking performance is quite a bit better, though still worse than observed.

D. **Try the Parallel General Executive**

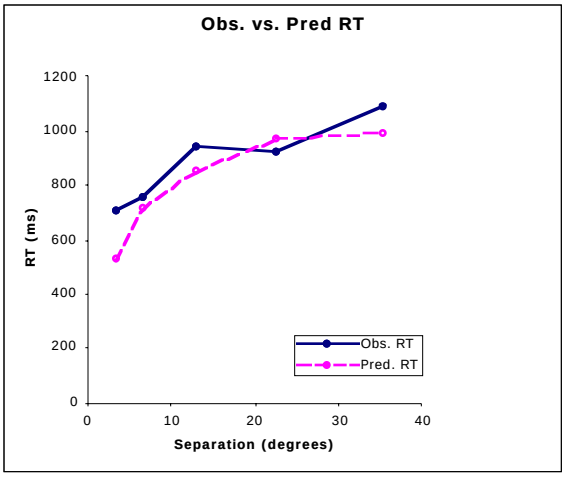
**Use "WStarterRules.prs" and "ParGenExStarter.prs" for rules:**

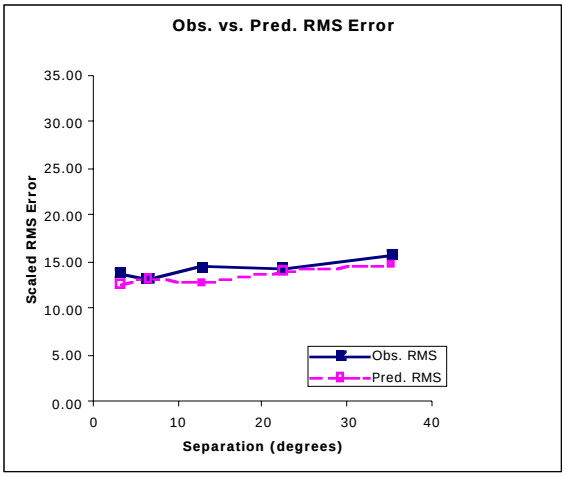
Use Startup and General Executive rules from ParGenExStarter.
Paste in and modify separate task rules from WStarterRules:

- Add Request/Has/Release items to interact with general executive.

Debug and explore with single-cycle running, then with a few trials.
Then do full data collection run and plot results.

**My results:**




Both RT and Tracking Error are in the right ballpark.
Tracking task was able to make some additional tracking motions!

III. **Discussion**

A. **What results did you obtain?**

B. **Modeling problems?**

C. **General comments?**