# Structured representations of human robot collaborative action

**Aaron F. Bobick**

(but really Kelsey P. Hawkins, Shray Bansal, and Nam Vo)

Center for Robotics and Intelligent Machines
School of Interactive Computing
Georgia Tech

# Modeling structured activity to support human-robot collaboration in the presence of task and sensor uncertainty

**Aaron F. Bobick**

(but really Kelsey P. Hawkins, Shray Bansal, and Nam Vo)

Center for Robotics and Intelligent Machines
School of Interactive Computing
Georgia Tech

Robotics@GT & Intelligent Machines

# Modeling structured activity in the presence of task and sensor uncertainty to support human-robot collaboration

**Aaron F. Bobick**

(but really Kelsey P. Hawkins, Shray Bansal, and Nam Vo)

Center for Robotics and Intelligent Machines
School of Interactive Computing
Georgia Tech

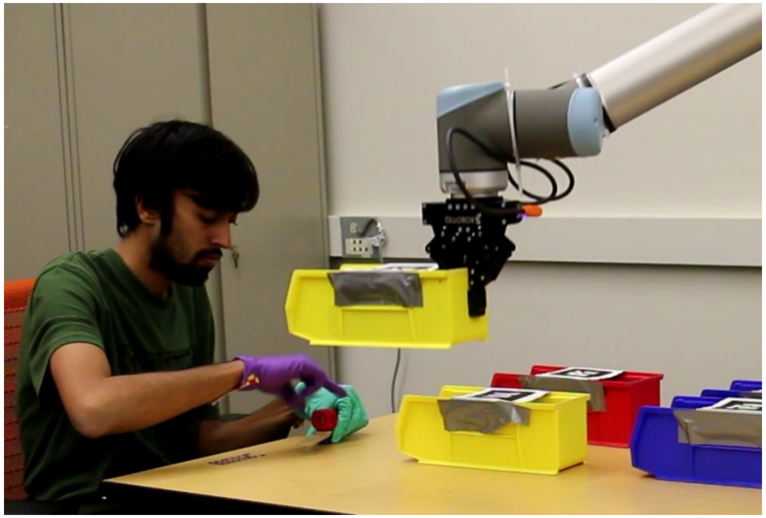
Robotics@GT & Intelligent Machines

# Anticipating the actions of humans

- Goal: Anticipate the actions of humans such that a robot can anticipate the needs of the human to provide assistance when needed – no waiting.

A challenge because:

1. Human collaborator doesn't do same thing every time, even in assembly situations
2. The rate at which they do it varies
3. Perception is (usually) an uncertain business
4. Robots take time to do things *(way too much time)*



# Integrate over task and perceptual uncertainty

## Our method:

1. Compile **structured representations** of activity into **probabilistic** system for reasoning about task and timing
   - *The variables of interest: the stop and end time of* ***sub-actions***
2. Learn from (very small amounts of) data:
   - ***Duration models*** *of sub-actions*
   - *Likelihood of branches in activity*
   - ***Perceptual detectors*** *that encode (noisy) information about the human performance (or start and end) of actions*
3. At **every time step**, perform **inference** on ***all*** actions.
4. Make **plans** based upon probabilistic assessment of what actions will be done and when  Minimize an HRI cost.

# Integrate over task and perceptual uncertainty

## Our method:

1. Compile **structured representations** of activity into **probabilistic** system for reasoning about task and timing
   - *The variables of interest: the stop and end time of **sub-actions***
2. Learn from (very small amounts of) data:
   - ***Duration models** of sub-actions*
   - *Likelihood of branches in activity*
   - ***Perceptual detectors** that encode (noisy) information about the human performance (or start and end) of actions*
3. At **every time step**, perform **inference** on ***all*** actions.
4. Make **plans** based upon probabilistic assessment of what actions will be done and when Minimize an HRI cost.

# Integrate over task and perceptual uncertainty

## Our method:

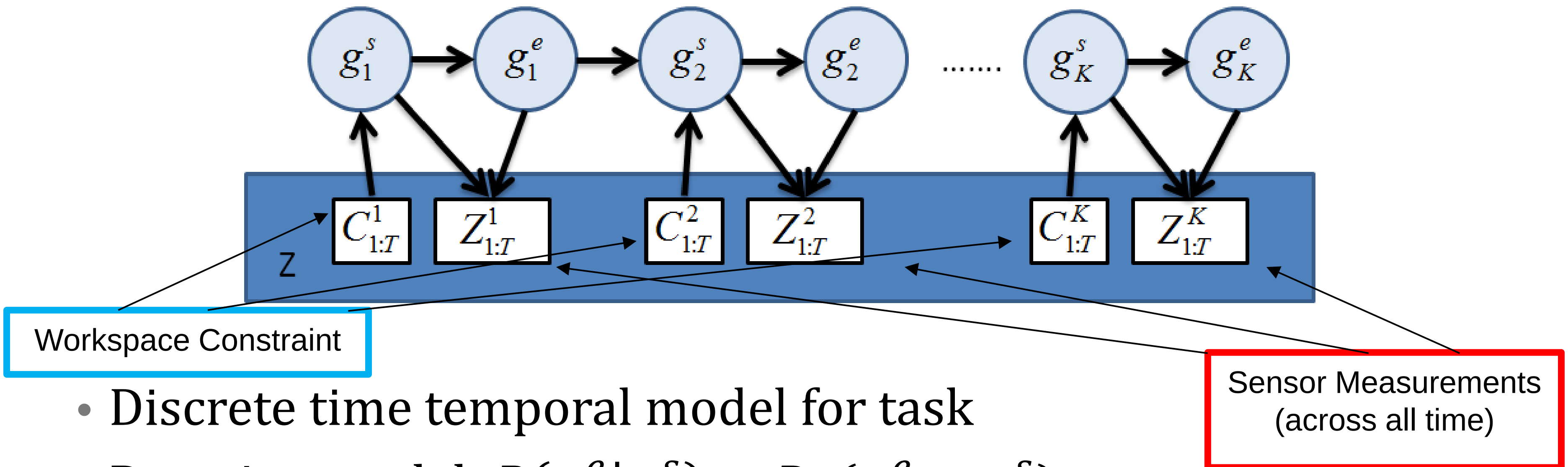
1. Compile **structured representations** of activity into **probabilistic** system for reasoning about task and timing
   - *The variables of interest: the stop and end time of **sub-actions***
2. Learn from (very small amounts of) data:
   - ***Duration models** of sub-actions*
   - *Likelihood of branches in activity*
   - ***Perceptual detectors** that encode (noisy) information about the human performance (or start and end) of actions*
3. At **every time step**, perform **inference** on ***all*** actions.
4. Make **plans** based upon probabilistic assessment of what actions will be done and when Minimize an HRI cost.

# Integrate over task and perceptual uncertainty

Our method:

1. Compile **structured representations** of activity into **probabilistic** system for reasoning about task and timing
   - *The variables of interest: the stop and end time of* ***sub-actions***
2. Learn from(very small amounts of) data:
   - ***Duration models*** *of sub-actions*
   - *Likelihood of branches in activity*
   - ***Perceptual detectors*** *that encode (noisy) information about the human performance (or start and end) of actions*
3. At **every time step**, perform **inference** on ***all*** actions.
4. Make **plans** based upon probabilistic assessment of what actions will be done and when Minimize an HRI cost.

# Sequential model *(Humanoids 2013)*

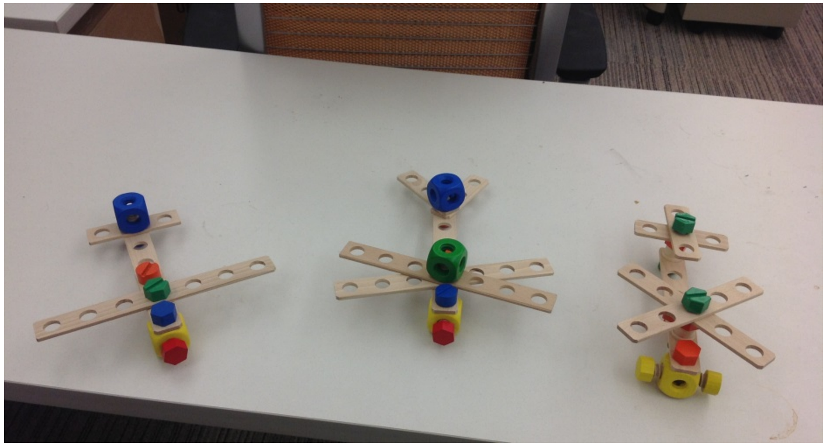


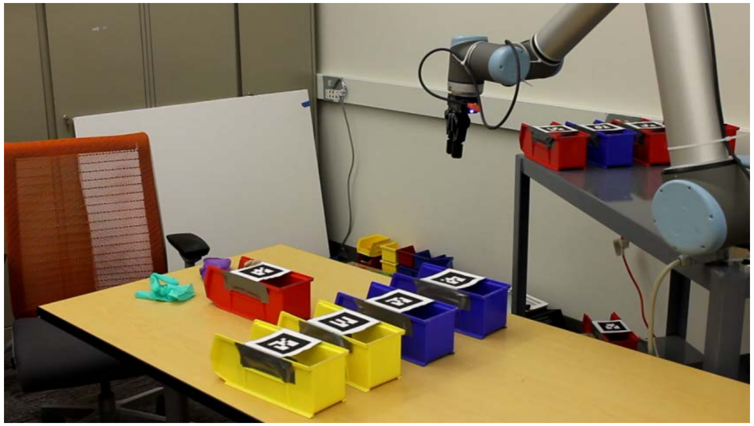

- Discrete time temporal model for task
- Duration model: $P(g_k^e|g_k^s) \propto D_k(g_k^e - g_k^s)$
- Sensor model: $P\left(Z^k|g_k^s, g_k^e\right)_{\{1:T\}}$ (non-informative future)
- Inference model (basic chain):

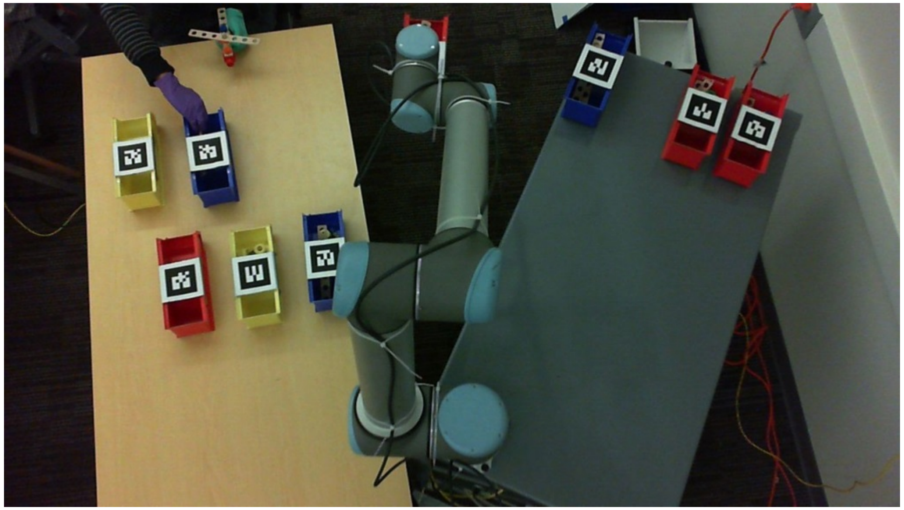
$$P(g,Z) = \prod_{k=1}^{K} P(g_k^s \mid g_{k-1}^e) P(g_k^e \mid g_k^s) P(Z^k \mid g_k^s, g_k^e)$$

- At ***every moment in time*** can infer the ***distribution*** **of** ***all*** the sub-tasks start and end times

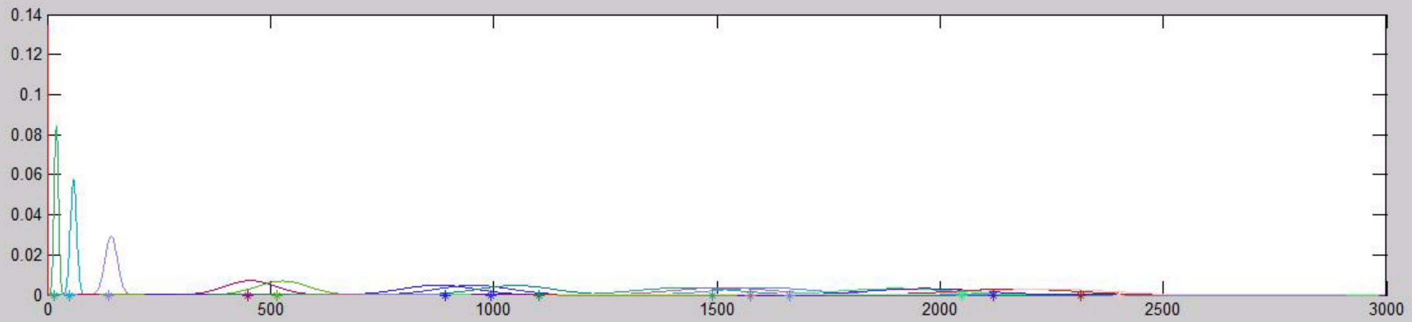
# Our domain

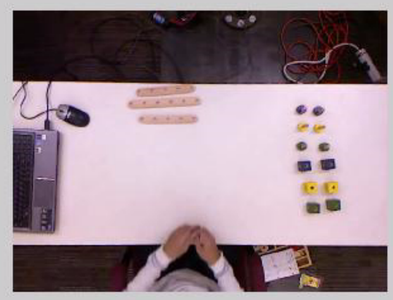






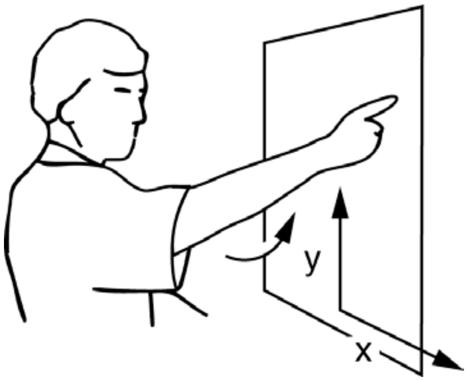
# Illustration of inference



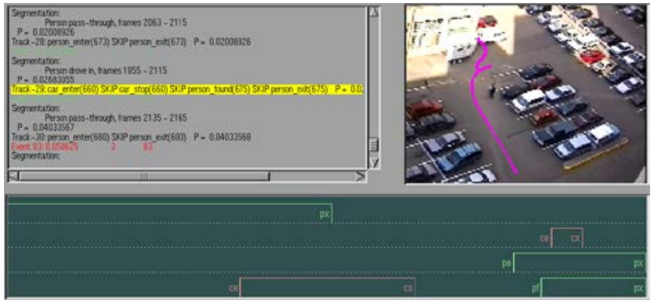



# A very brief history of some of our computer vision work…

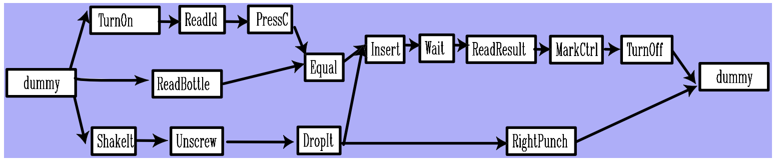
- *Parametric HMMs* for "structured" gesture recognition
  - Coupled parametric modeling with ***graphical model inference***




- *Stochastic Context Free Grammar* based representation and parsing
  - ***Richly expressive for activity description***
  - Easy to build higher level activity from reused low level vocabulary.



- *P-Net (Propagation nets)*
  - Focused on ***intervals***
  - Specify the structure – with some annotation can ***learn detectors and triggering probabilities***



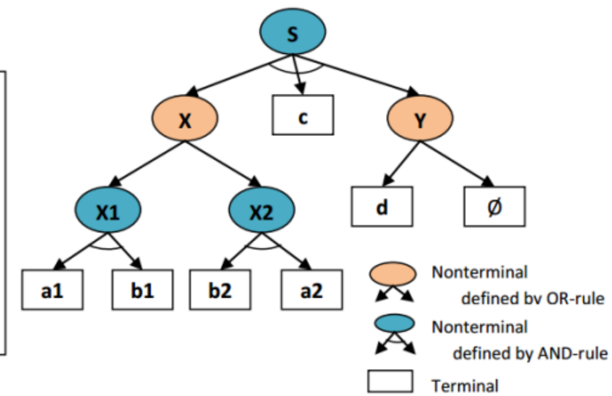
# Grammars: More interesting task descriptions

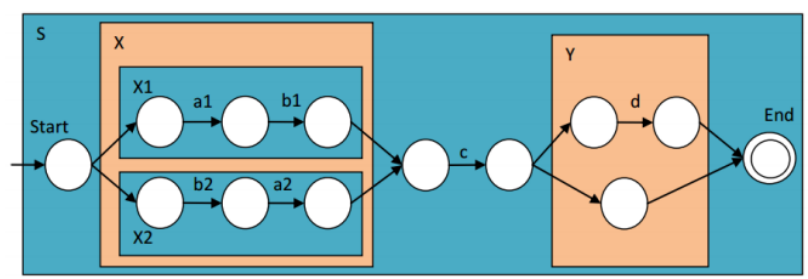
- First do "**a**" and "**b**" in any order, then do "**c**" and optionally then do "**d**"
- Can be written as a (trivial) grammar:

  $S \rightarrow (\mathbf{a\ b} \mid \mathbf{b\ a})\ \mathbf{c}\ (\mathbf{d} \mid \emptyset)$

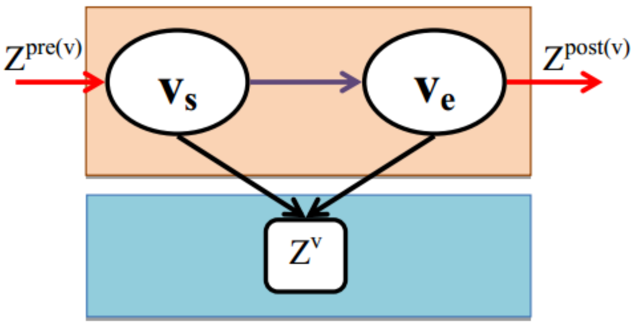
- An AND-OR tree that expresses temporal ordering and selection

S $\rightarrow$ X c Y
X $\rightarrow$ X1 | X2
X1 $\rightarrow$ a1 b1
X2 $\rightarrow$ b2 a2
Y $\rightarrow$ d | Ø



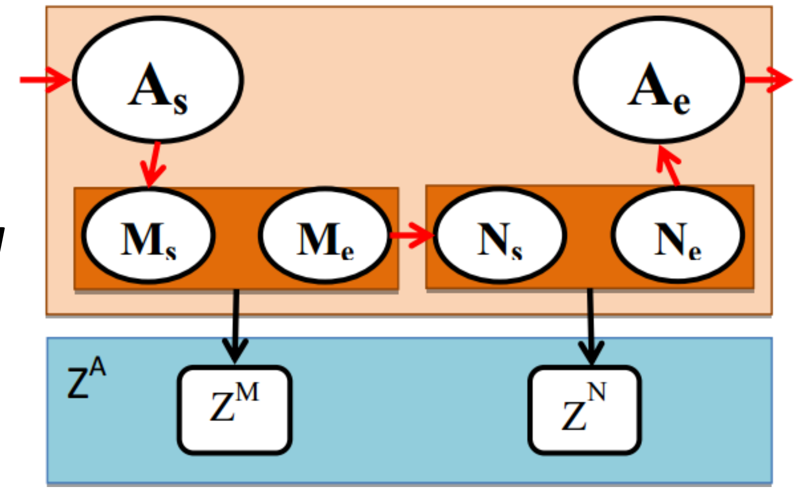



# From AND-OR to Bayes Networks (ICRA '2014)

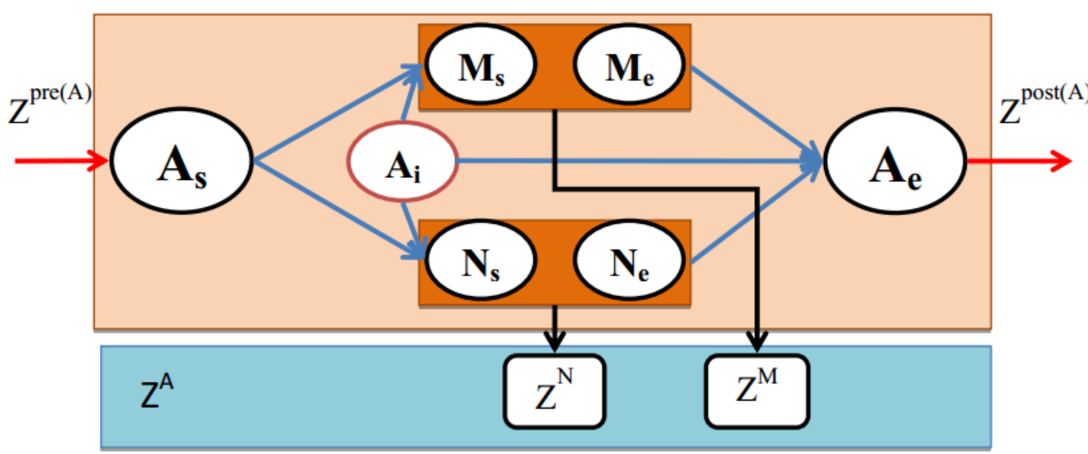



Primitive action **v**

AND: ***A -> MN***




OR: ***A -> M | N***



# Some gratuitous math…

**The input include:** (1) $P(\exists S)$: the prior probability of S happening, (2) $P(S_s|\exists S)$: The prior probability of the start of S, (3) $P(Z^{end}|S_e, \exists S)$: the likelihood representing the constraint on the end of S, and (4) the CPT $P(A_e|A_s)$ and $P(Z^A|A_s, A_e)$ for all primitive action A (recall that our random variables have discrete values between 1 and T. The special value $A_s = A_e = -1$ means $\exists A$, the case where the action A happens)

**Step 1, Forward phase:** Given $P(A_s, Z^{pre(A)}|\exists A)$, one can compute $P(A_e, Z^{pre(A),A}|\exists A)$ for every action A (where $Z^{pre(A)}$ stands for the observation of all actions happening before A). If A is a primitive action, then compute the joint $P(A_s, A_e, Z^{pre(A),A}|\exists A)$ and perform marginalization. If A is defined as M AND N, then recursively compute $P(M_e, Z^{pre(A),M}|\exists M)$ and $P(N_e, Z^{pre(A),M,N}|\exists N)$ then we have the distribution of $A_e$ the same as $N_e$. On the other hand if A is defined as M OR N, then the distribution of $A_e$ will be weighted combination of $M_e$ and $N_e$ according to equation 3.

The forward process starts with $P(S_s|\exists S)$ and recursively compute $P(A_s, Z^{pre(A)}|\exists A)$, $P(A_e, Z^{pre(A),A}|\exists A)$ for every action A

**Step 2, Backward phase:** similarly, this process starts with $P(Z^{end}|S_e, \exists S)$ and recursively compute $P(Z^{post(A)}|A_e, \exists A)$, $P(Z^{A,post(A)}|A_s, \exists A)$ for every action A (here $Z^{post(A)}$ stands for observation of all actions happening after A).

**Step 3, compute the posteriors:** this is done simply by multiplying the forward and backward messages, we obtain $P(A_s, Z|\exists A)$ and $P(A_e, Z|\exists A)$ for every action A. Additionally we can have $P(Z) = \sum_{t>0} P(S_s = t, Z)$

**Step 4, compute the posterior probabilities of an action happening:** starting with $P(\exists S|Z) = P(\exists S) = 1$, evaluate $P(\exists A|Z)$ for every symbol A recursively.

Given S is defined as A AND B, then $P(\exists A|Z) = P(\exists B|Z) = P(\exists S|Z)$.

Given S is defined as A OR B, one can compute (apply similar formulas for B):

$$P(\exists A|Z) = P(\exists S|Z)\frac{P(\exists A, Z|\exists S)}{P(\exists A, Z|\exists S) + P(\exists B, Z|\exists S)} \quad (1)$$

where $P(\exists A, Z|\exists S)$ can be calculated:

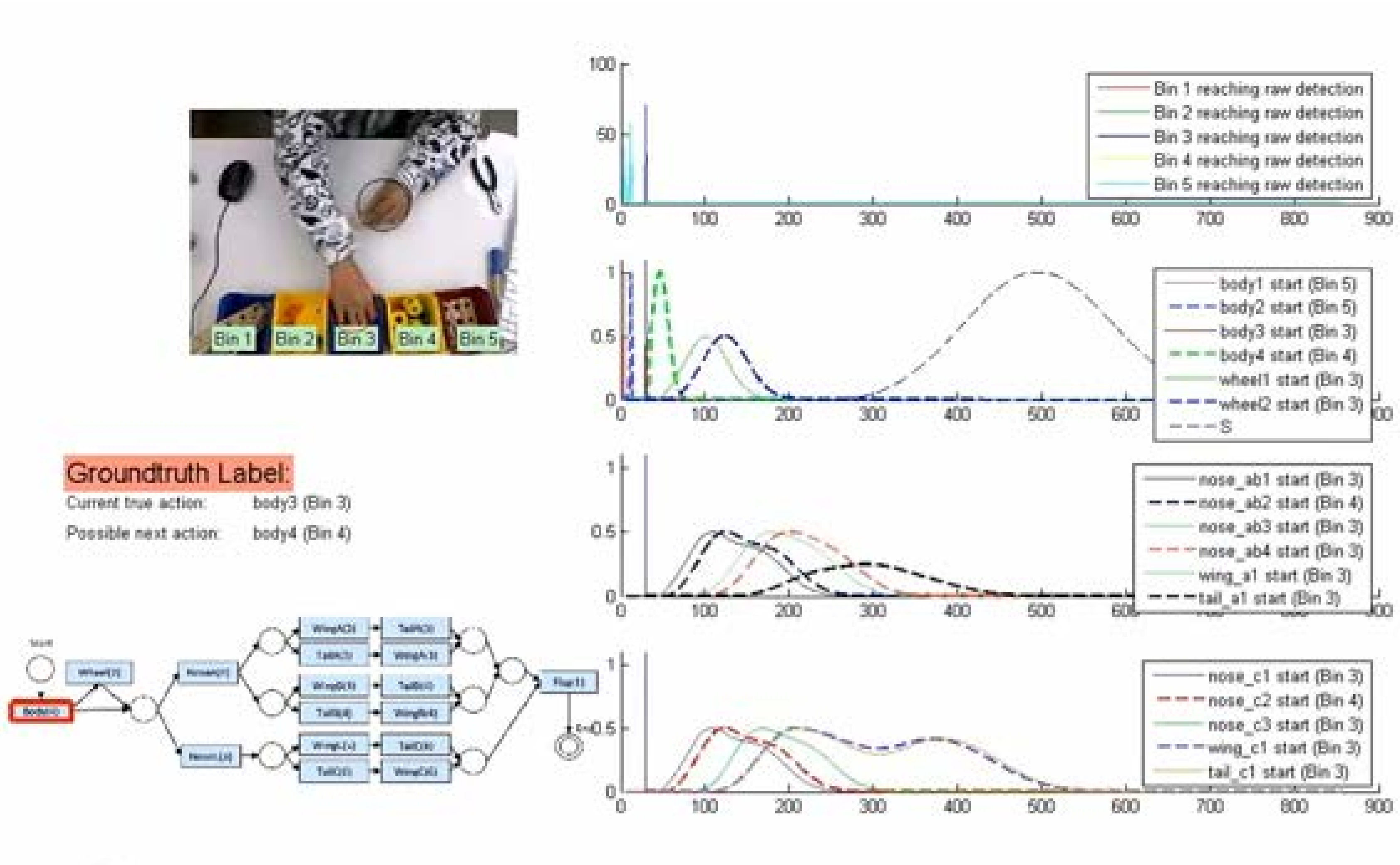
$$P(\exists A, Z|\exists S) = P(\exists A|\exists S)\sum_{t>0} P(A_e = t, Z|\exists A) \quad (2)$$

**Output:** the probability of action A happening $P(\exists A|Z)$, and if that the case, the distribution of the start and end $P(A_s, Z|\exists A)$, $P(A_e, Z|\exists A)$. We can compute:

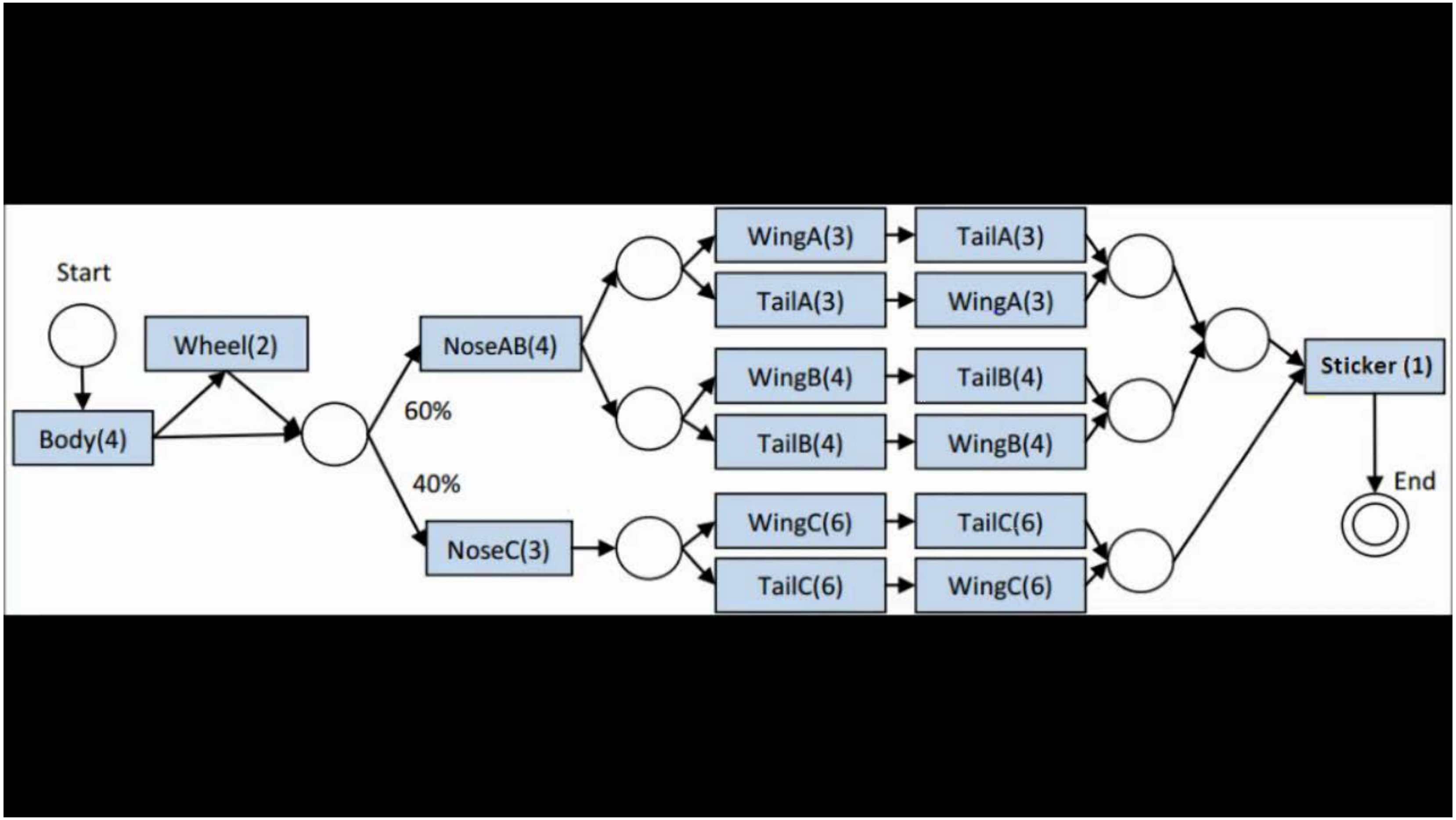
$$P(A_s|Z) = P(\exists A|Z)\frac{P(A_s, Z|\exists A)}{\sum_{t>0} P(A_s = t, Z|\exists A)} \quad (3)$$

for values between 1 and T. Note that $P(A_s = -1|Z) = P(!A|Z) = 1 - P(\exists A|Z)$.

# Evolving prediction uncertainty



# Parsing (only) video *(and more CVPR 2014)*



# Back to robotics…

# Given timing distributions, we need a plan

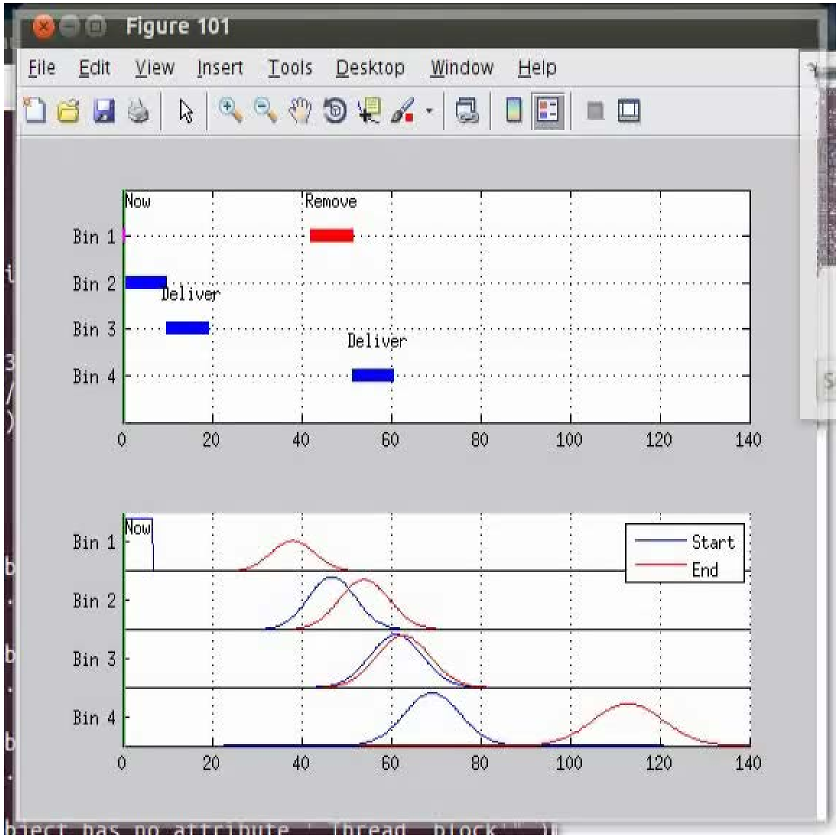
- Two sources of "cost": remove a bin early, deliver a bin late. Can be condensed to function of individual wait times:

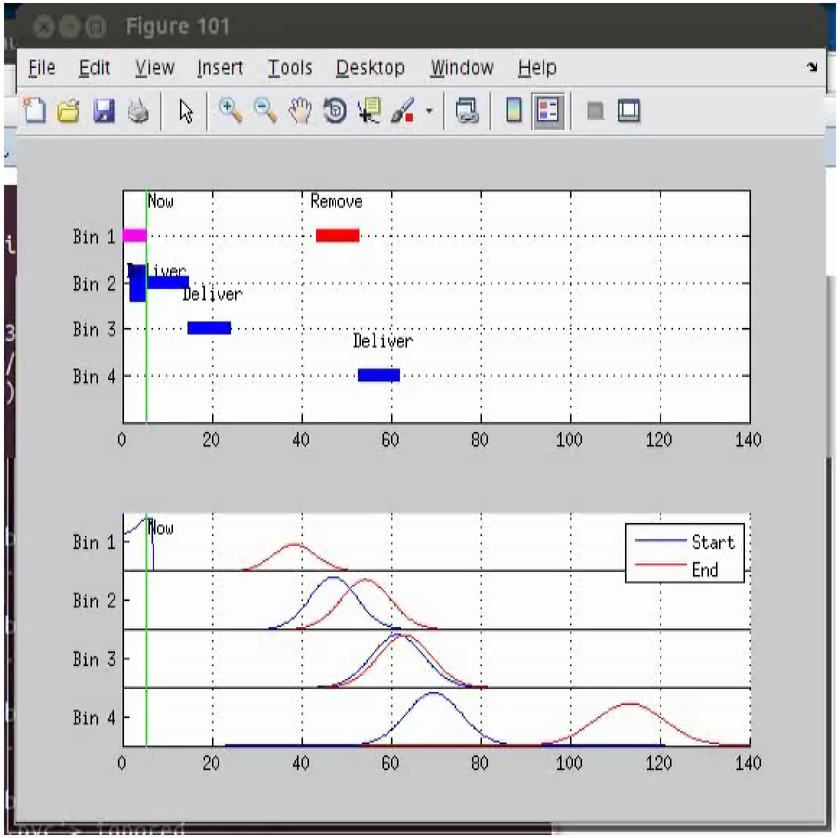
$$C = \sum_i \Psi(w_i)$$

where *i* is for each time the human needs to wait, *wi* is the amount of wait time *i*, and Ψ is sum HRI determined function we used quadratic) Note this is not necessarily total execution time.

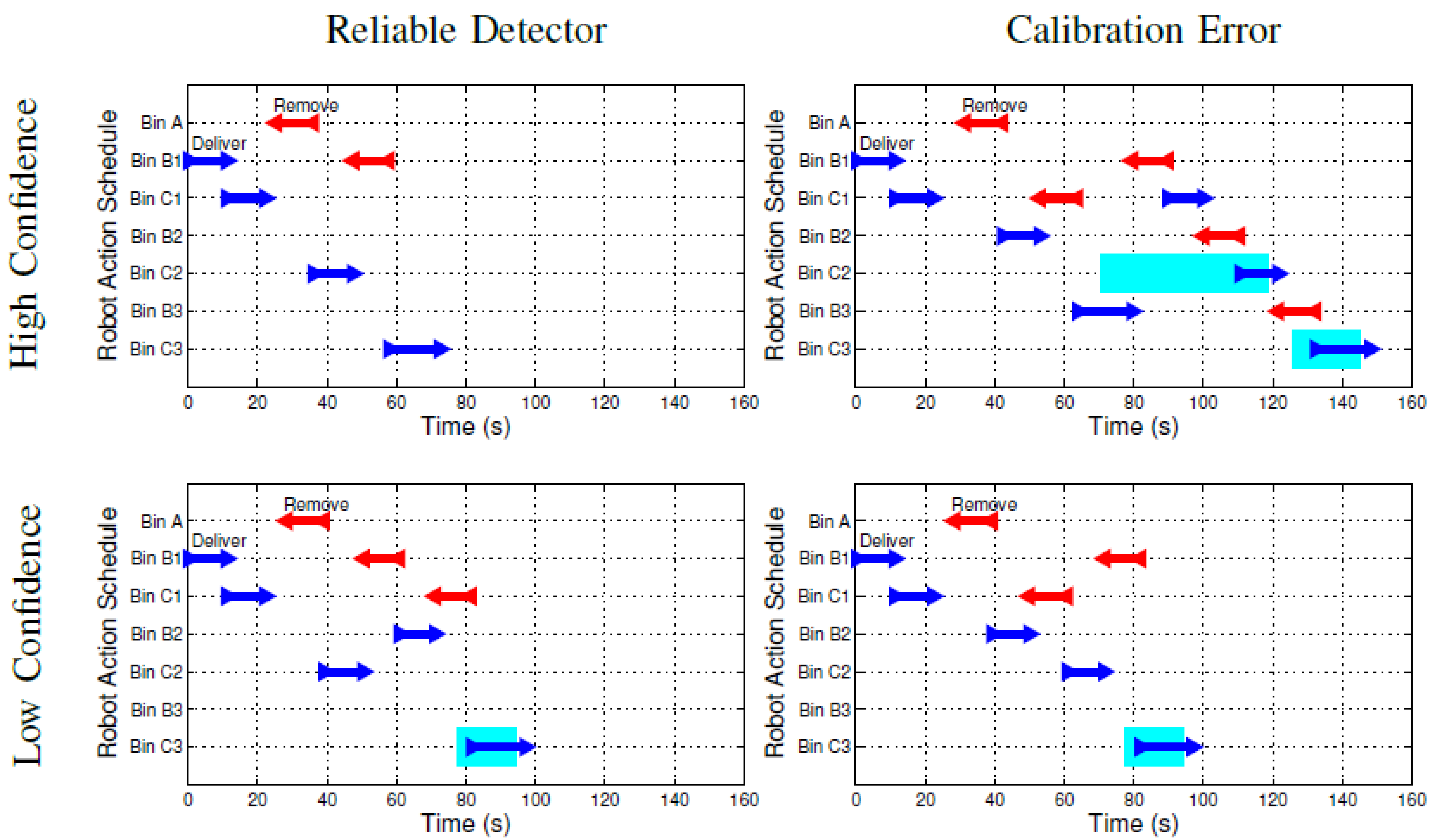
- Planning is a heuristic over the independently considered intervals.

# Planning in action



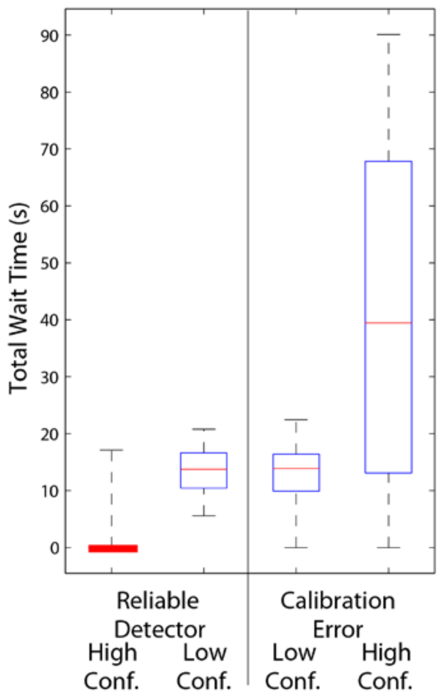
# Certainty of belief affects plan



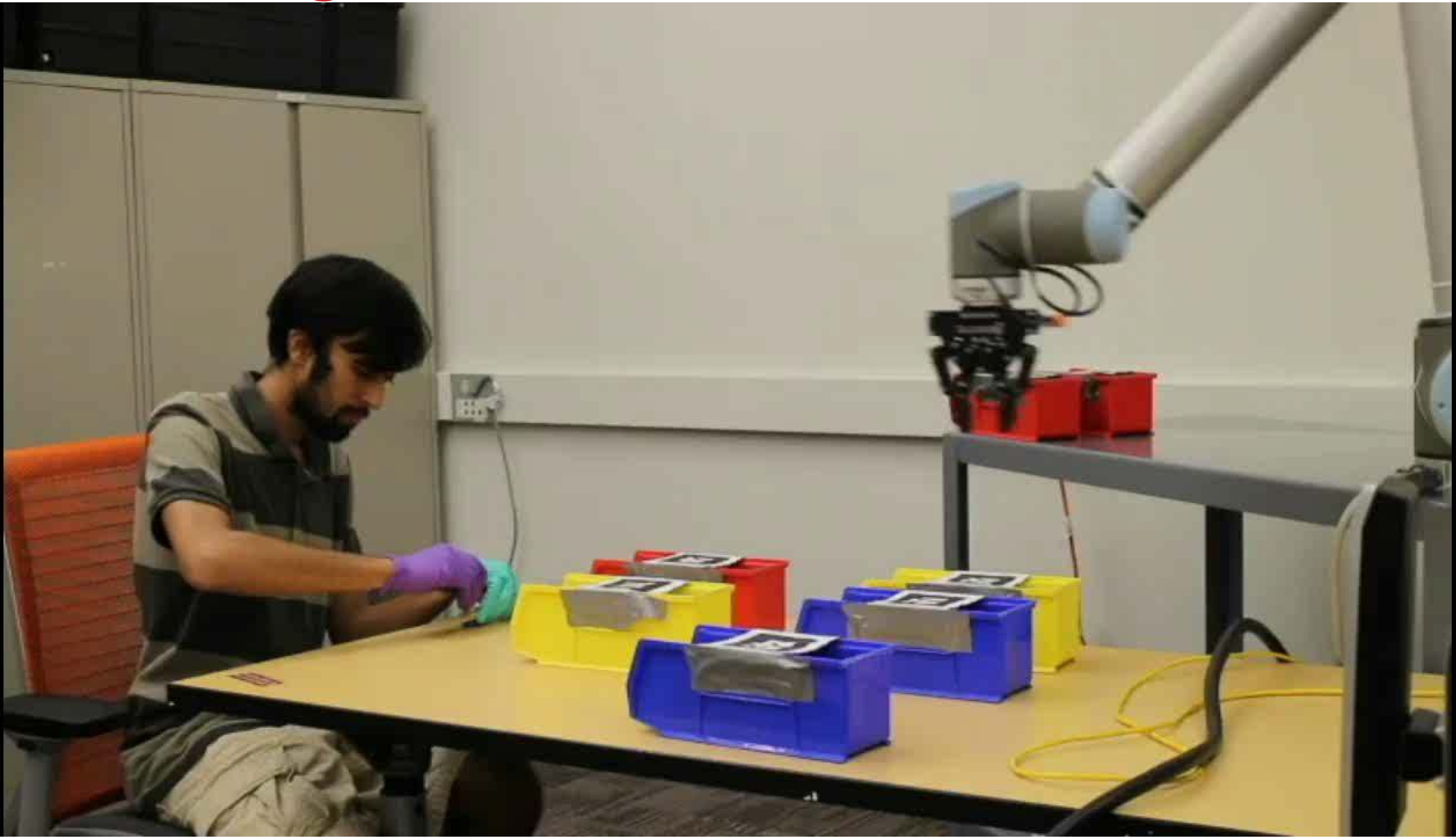
# Simulation

- Can explicitly vary *expected uncertainty* with *actual uncertainty* (or variability)

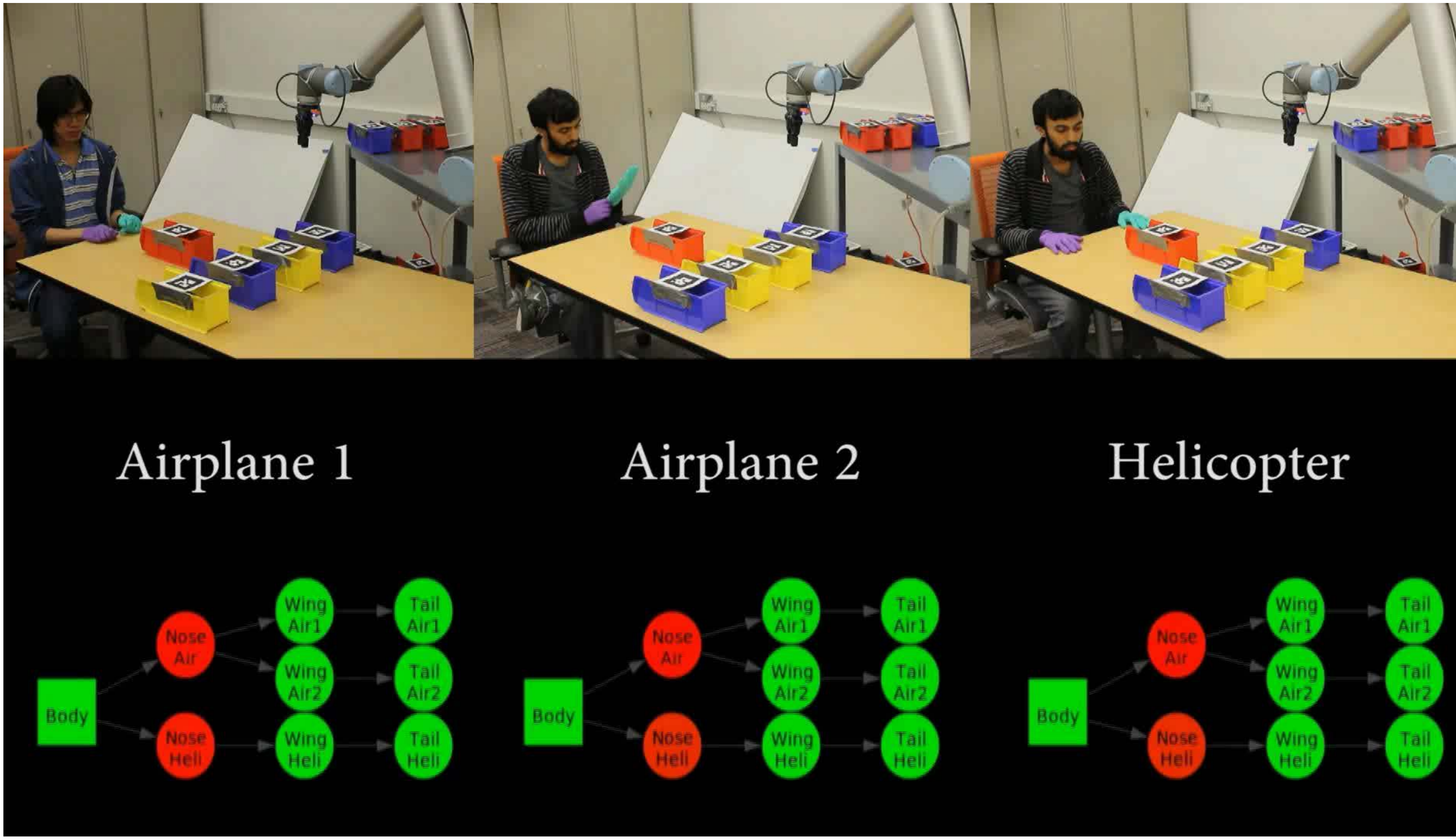
| | Low Noise | | Detector Noise | |
|---|---|---|---|---|
| Model | Task Execution Time | Total Waiting Time | Task Execution Time | Total Waiting Time |
| Low Noise Expectation | 102.19 | 1.54 | 125.34 | 23.56 |
| High Noise Expectation | 108.93 | 7.78 | 111.84 | 10.22 |



# Working with the robot



# Task indeterminacy



# Conclusions

- Three sources of uncertainty in a robot needing to anticipate human action:
  - Variation in what the human does
  - Variation in how they do it (speed)
  - Uncertainty in perceptual sensing
- Main idea: reason about the likely timing of the start and end of sub-actions given evidence observed so far.
- Automatic conversion from task-level specification
  - Learn probability models from limited data *(???)*
- (Some) Open problems: learning the grammar, detection of being “off-task”