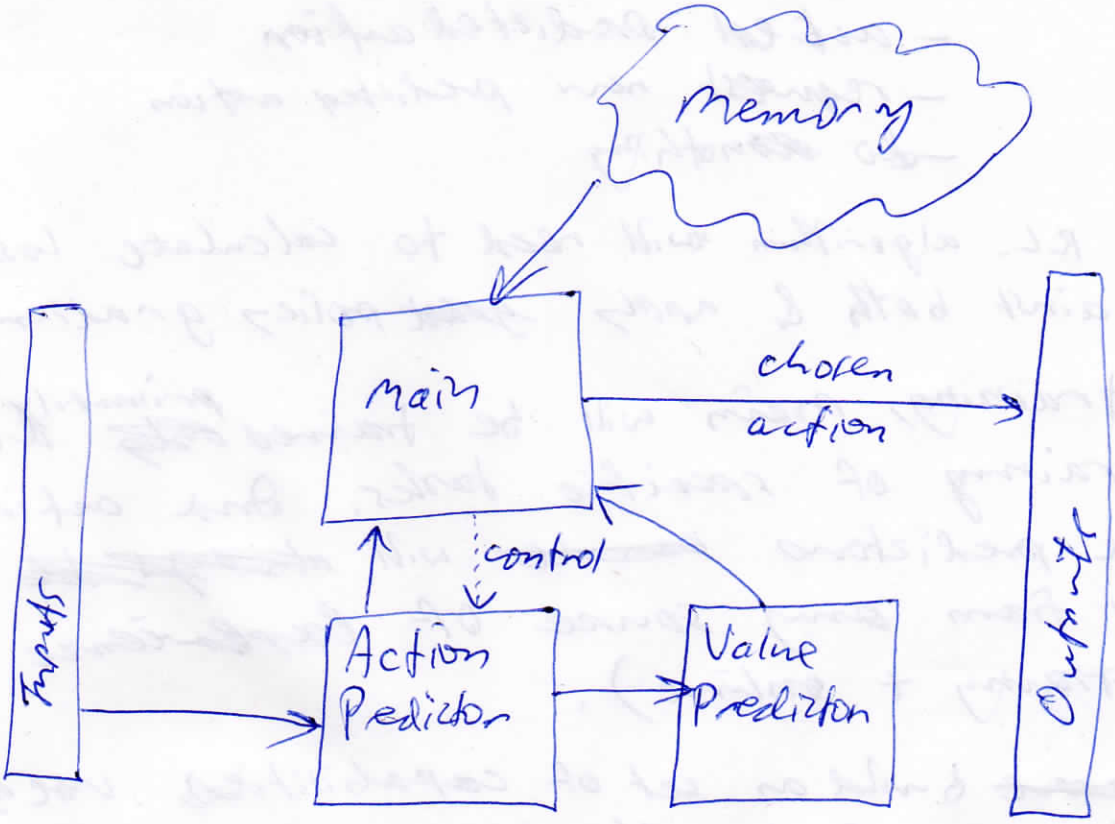
Deconstructed R.L. Agent as basic for AGI



- Main receives ~~a~~ suggested action from Action predictor. Action representation is likely ~~be~~ high-level & continuous.
- Main also receives predicted value of action, and uses that to decide whether to accept action. If it accepts action then it is sent to outputs. Otherwise no action is taken.
- Main can alternatively choose to request a new action prediction. The trigger may be because the ~~exp~~ predicted value is too low, or because of ~~too~~ memory of ~~that sort~~ the current predicted action already being attempted & not working. The action predictor could be used in a continuous stochastic way, and thus merely requesting a new prediction could produce a new action.
  In a more advanced form, some sort of inhibition of the previous prediction(s) would be applied.

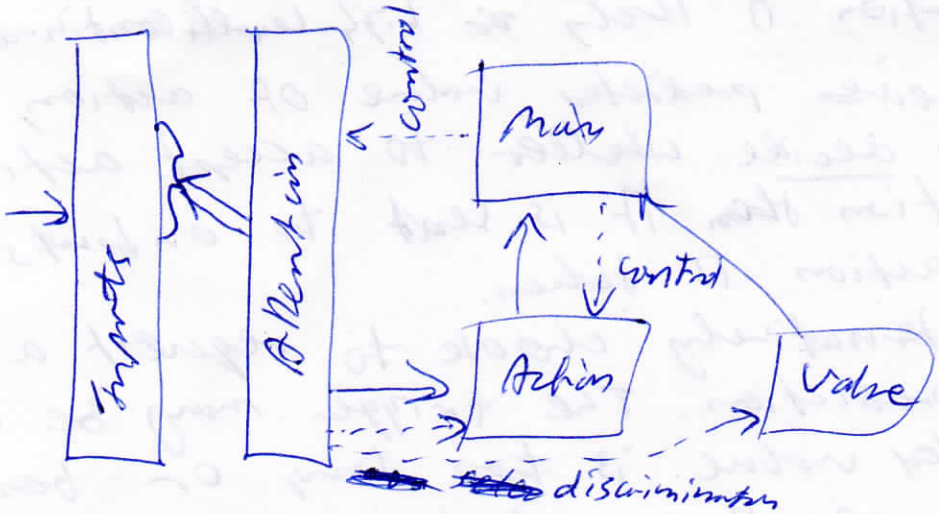
2/

- main now has actions itself. These are simple categorical actions:
  - accept predicted action
  - request new predicted action
  - do nothing.

  So RL algorithm will need to calculate loss against both & apply policy gradients.

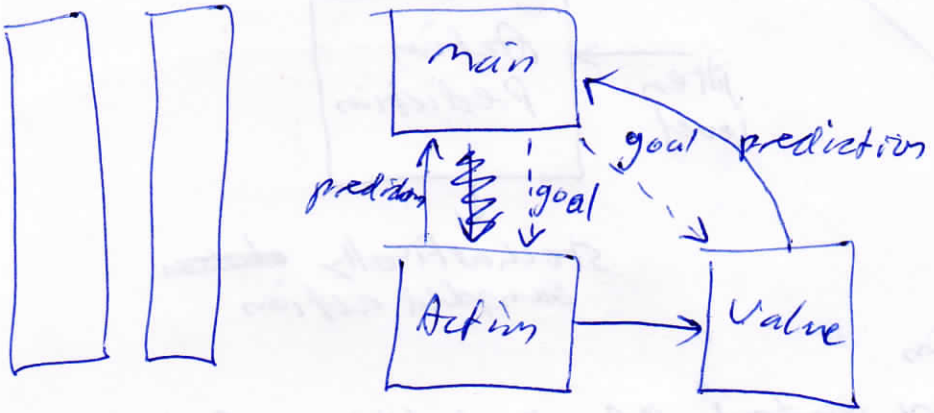
- For training, main will be trained primarily during pre-training of specific tasks. And action & value predictors will be trained from any source of experience (pre-train + online).

- To build on set of capabilities, use attention to act as a discriminator or variable that selects particular sub-networks.



  - Action & value networks have 'input + attention' as inputs so that they can operate against different scenarios ← Ideally needs catastrophic forgetting protection.
  - Main can control attention.

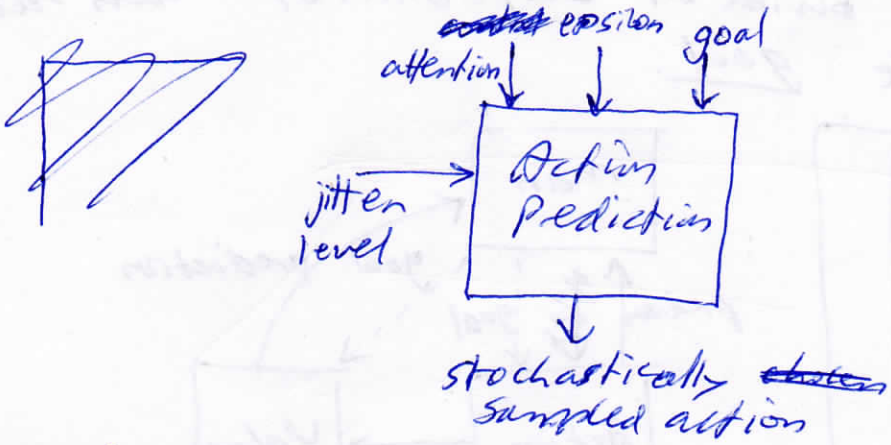
- To further build on capabilities, Main needs to control the goal.



- Main can now learn its own higher-level strategies for driving the system. As new capabilities are added it will have more options for the sources of predicted actions (eg: modifiable action prediction blocks, or choose different action groups through goal selection). And it will develop different strategies for exploiting possibilities.

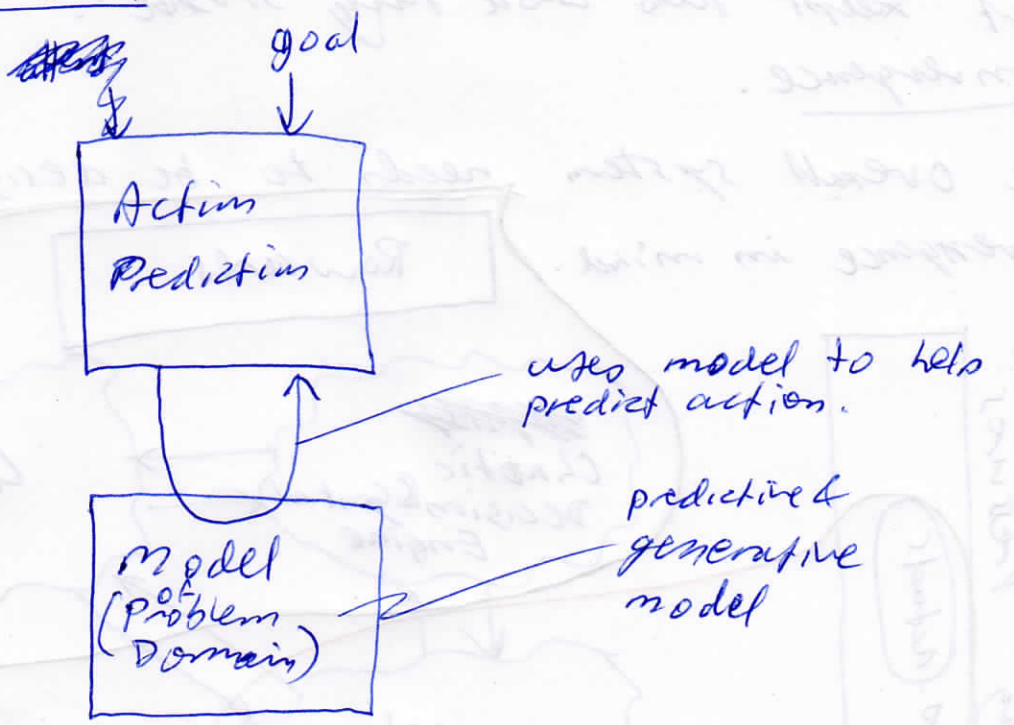
- Ideally also need a modelling capability to build up novel strategies, to build theories of external systems to aid in action/value prediction, and to discover & control its own learning agenda.

4/ • Emulation baby "jitter learning" & stochastic experimentation:



- Treat min output as probabilities & take a stochastic sample.
- Let main directly control (see pg7.) experimentation vs. exploit through control of epsilon.
  (choose completely random action output with probability ε)
  or better yet: through the level of noise
  (std. devs increased according to level of requested noise).
  ↳ this makes the agent occasionally make small mistakes when it knows what it's doing, but can also experiment widely when it doesn't know.
- Lastly, human babies learn initial models of their limbs through enforced jitter, that acts as a source of initial learning.

- Combine with model



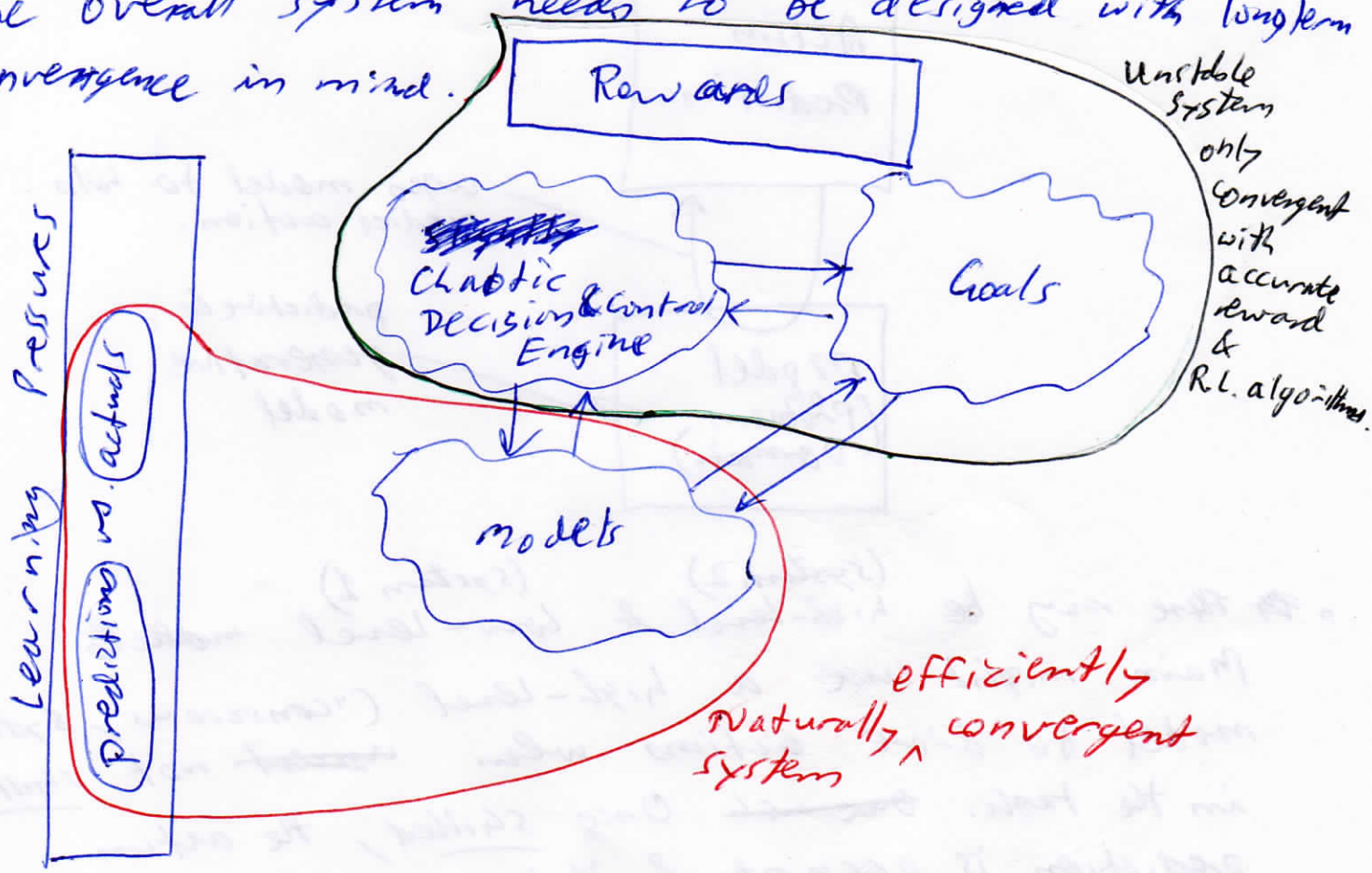
- There may be (system 2) high-level & (system 1) low-level models. Brain might use a high-level ("conscious", system 2) model to drive actions when not skilled in the task. Once skilled, the action prediction is accurate & that can be blindly trusted as system 1 thought, without using a system 2 model.

6/

- What keeps this whole thing stable?

A: Convergence.

The overall system needs to be designed with long term convergence in mind.

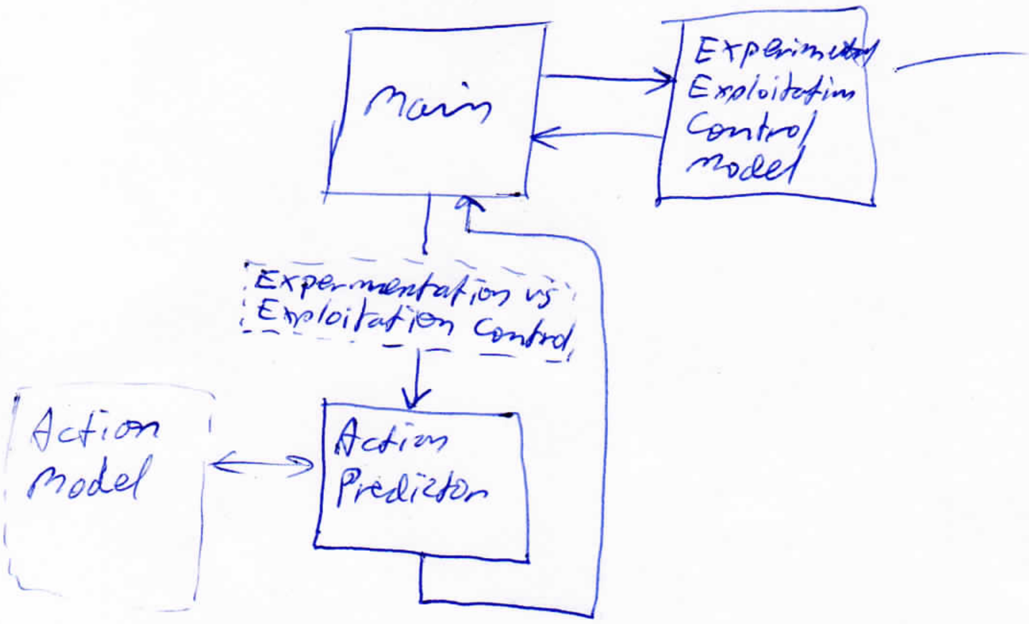


The naturally convergent model + prediction + actuals system is really important to keep the main decision engine in check

~~Making Self-Controlled Learning w0~~

Self-Controlled Epsilon/Noise Level (see p3 4).

To make this work, can't just rely on implicit pre-trained handling. Need explicit online handling of experimentation vs. exploitation control. It needs to learn a <u>model</u> that represents its benefits & behaviours.



Learning to use this model becomes one of many "strategies" that the agent can employ.

The E.E. control model will itself need to be learned over time. It is probably a bayesian inference model. And some sort of bootstrapping system will be needed to effectively bypass <u>executive control</u> during initial agent training when the E.E control model is too inaccurate to use.