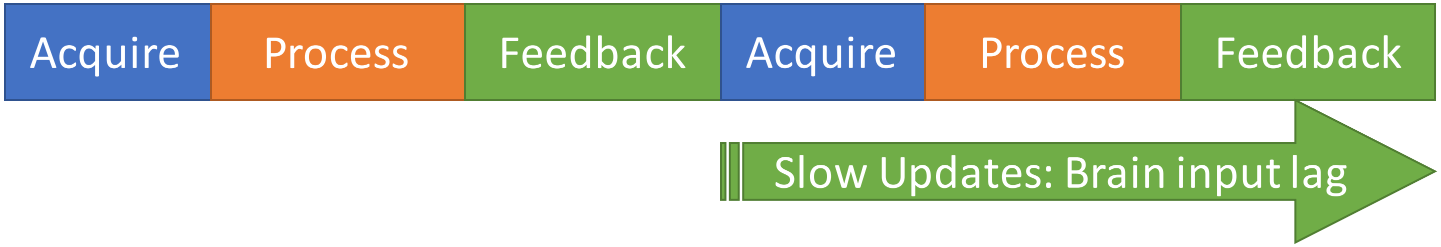
BCI the Wrong Way:
A single monolithic script

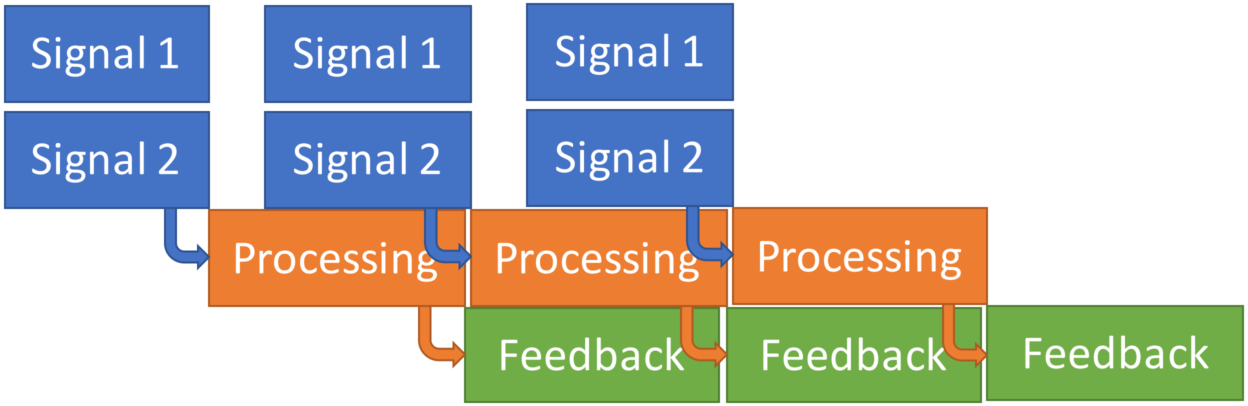
```
• while continue == True:
    • data = acquire_from_device(n_samples)
    • buffer = add_to_buffer(data)
    • filtered_data = filter(buffer)
    • features = extract_features(filtered_data)
    • control_signal = apply_model(features)
    • update_cursor_position(control_signal)
    • stim_events = update_stimulus()  # e.g. targets, cues
    • render_scene()
    • stream_to_file(data, stim_events)
```



Acquire | Process | Feedback | Acquire | Process | Feedback

Slow Updates: Brain input lag

BCI the Right Way:
Modular, Separate Processes

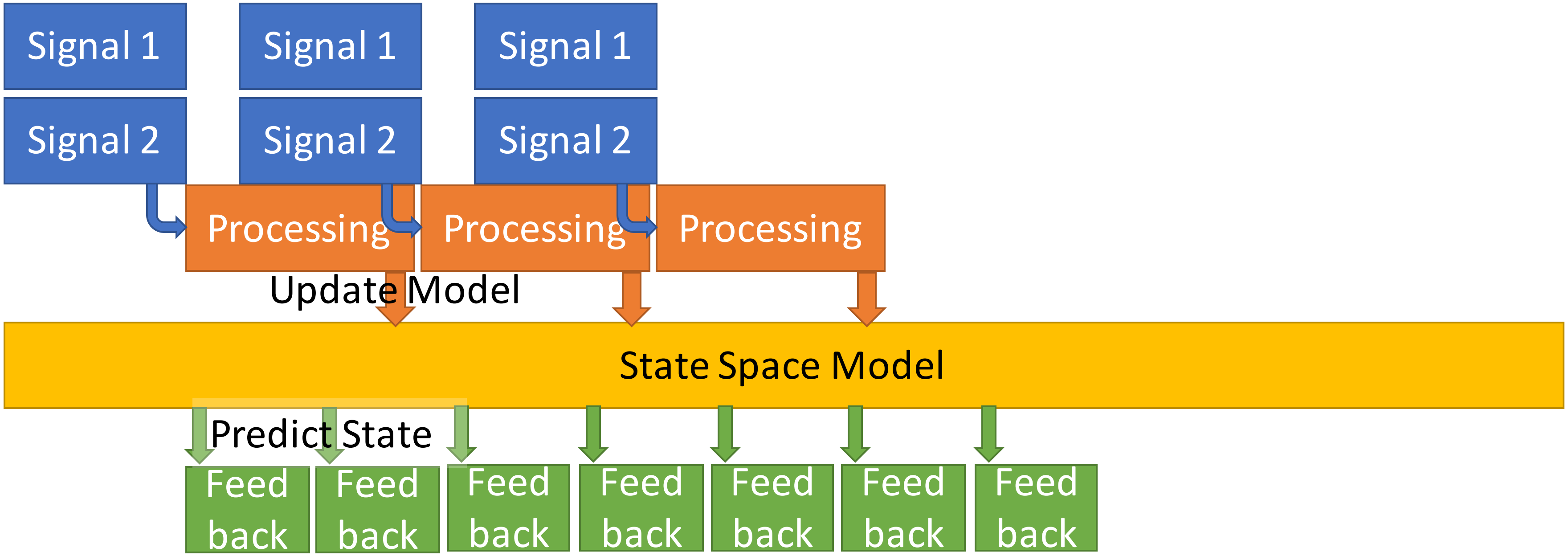


- Separate process for each signal acquisition.
  - (almost) Always has its own buffer or queue.
- Separate process for signal buffering and processing.
  - Fetches data from acquisition buffer or queue.
  - Processes signals, extracts features, updates model if necessary, applies model
- Separate process for stimulus presentation and feedback

Relative to monolithic script:

- Reduced time between feedback updates
- → Fewer samples → Slightly faster processing → Slightly reduced “brain input” lag

BCI the Better Way:
Decoupled Processing and Feedback



- When control signal follows dynamical rules
- State space model updated with new feature input.
- Model gets new timestamp on update.
- Feedback at time t + t_delta asks for state *prediction* (t_delta) from model.
  - t_delta can include feedback latency (e.g., display latency) if known.