Synchronizing Machine Learning Algorithms, Realtime Robotic Control and Simulated Environment with o80

Vincent Berenz, Felix Widmaier, Simon Guist, Bernhard Schölkopf and Dieter Büchler

o80 is a software for:
- managing inter-processes data streams exchange
- inter-processes synchronization

It is templated realtime safe C++, with automated generation of Python bindings.
Open source: https://github.com/intelligent-soft-robots/o80

Deployment
1. develop the classes for the driver (specifies input and output to robot) and the joints (information related to the robot state)
2. create the Python bindings at compile time
3. start the (realtime safe) o80 backend in Python
4. interact with the backend via a frontend (in Python)

Flexible command system

```python
# requesting backend to iterate (if bursting mode) used to retrieve data and to synchronize with the backend
```

Reading data and synchronizing

```python
# reading the latest observation
observation = frontend.latest()
iteration = observation.get_iteration()

# reading an observation from the past
observation = frontend.get_latest(10000)

# reading the latest 100 latest observations
observations = frontend.get_latest_observations(100)

# reading the latest observation
observation = frontend.get_next_obervations_since(iteration)

# waiting for future observations
while True:
    observation = frontend.wait_next_obervations_until(iteration + 100)

# waiting and reading commands
while True:
    iteration += 10
    observation = frontend.read(iteration)
    frontend.pulse()
```

Bursting mode

In bursting mode, the backend iterates upon requests by the frontend. Typically this is used for the control of simulated robot running accelerated time. This allows the simulation to synchronize with the control algorithm.

example:
The control algorithm computes the commands to be sent, then request the simulation to perform 10 iterations as fast as possible

```python
while True:
    observation = frontend.latest()
    # creating and sending commands
    frontend.burst(10)
```

https://ei.is.mpg.de