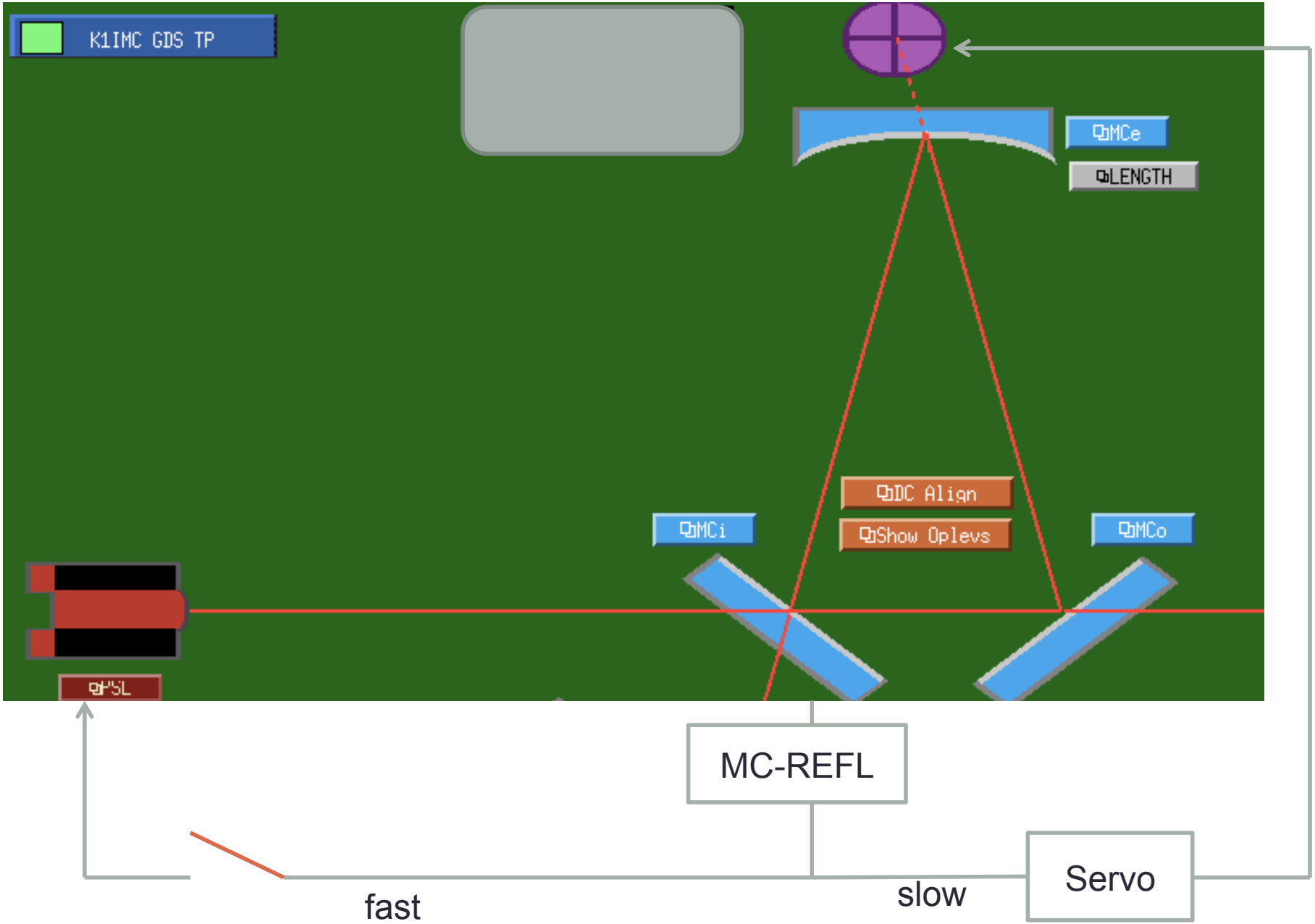
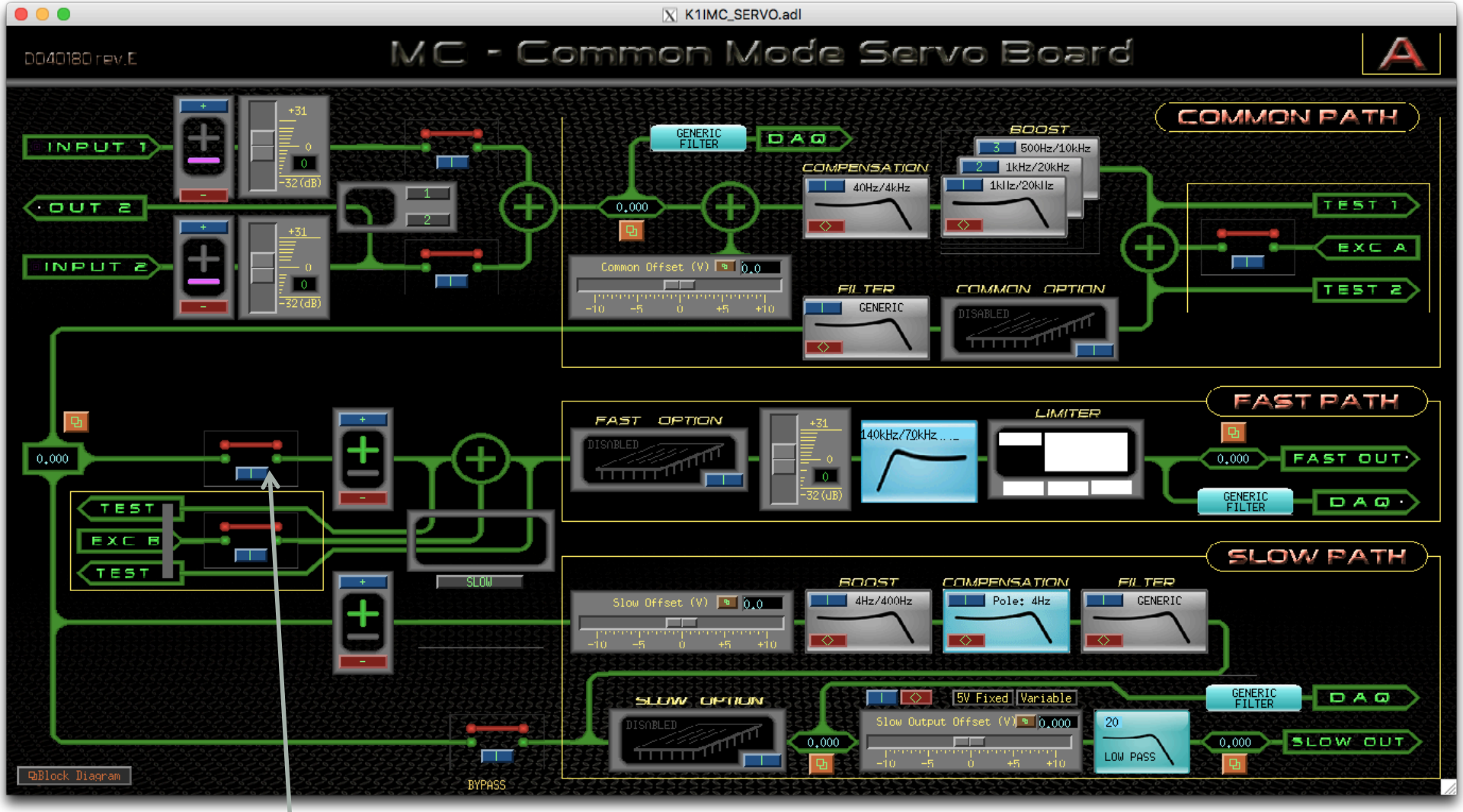
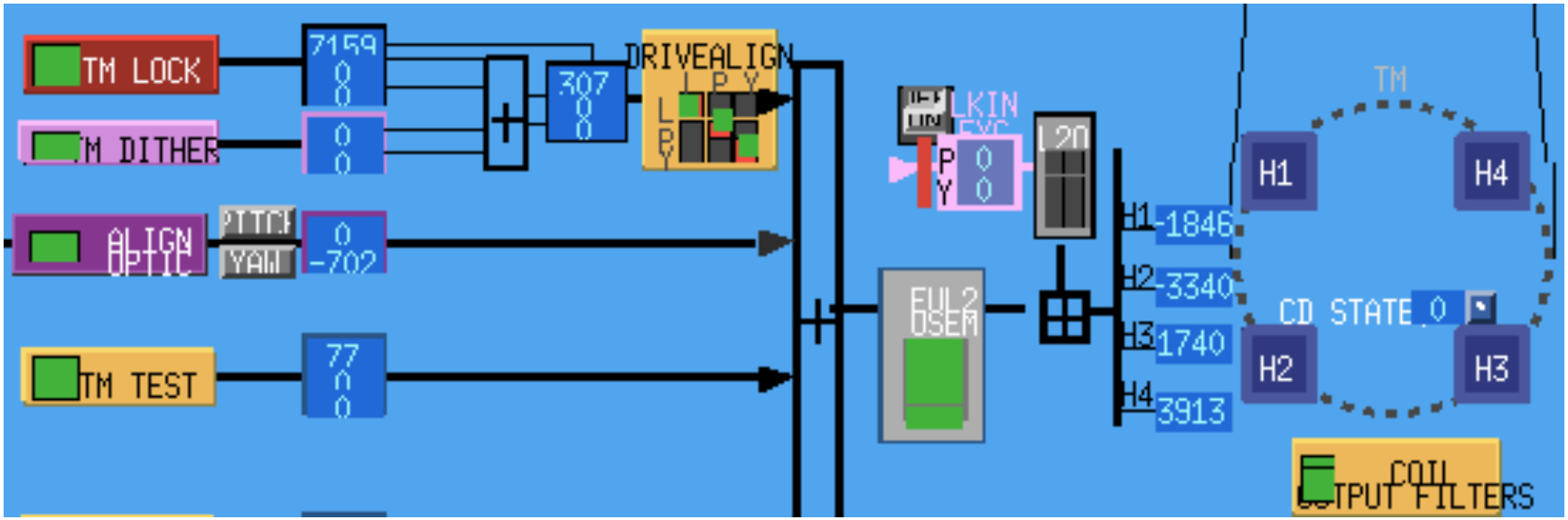
IMC LOCK

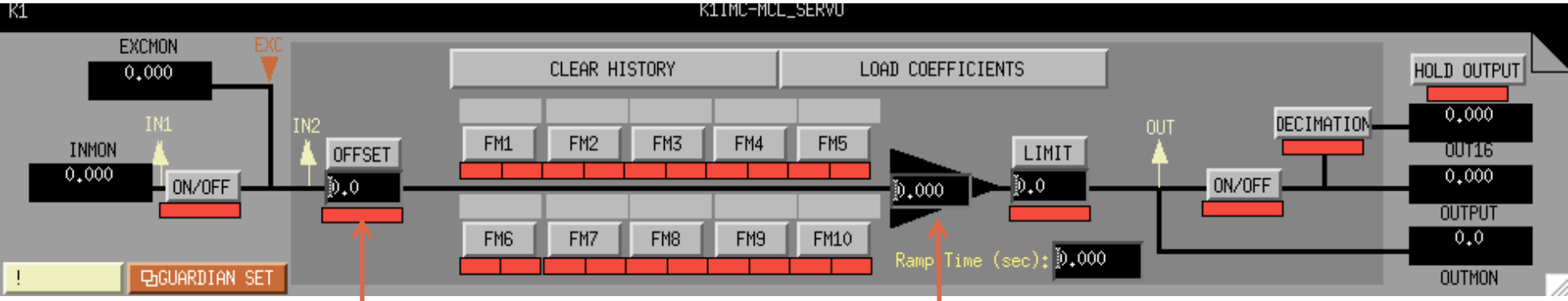
Nagaoka U.of Tech., Yukitsugu Sasaki


K1IMC GDS TP
MCe
LENGTH
DC Align
Show Oplevs
MCi
MCo
PSL
MC-REFL
Servo
fast
slow



IMC-REFL_SERVO_FASTEN



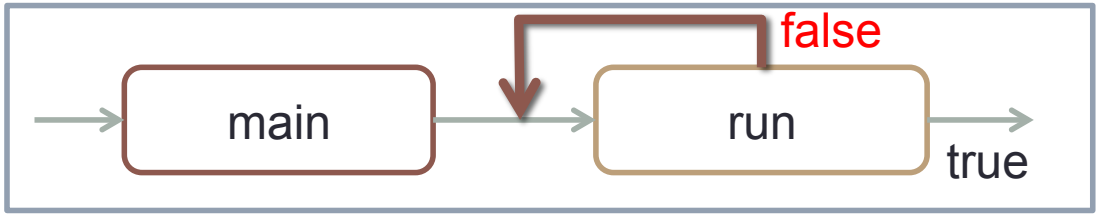


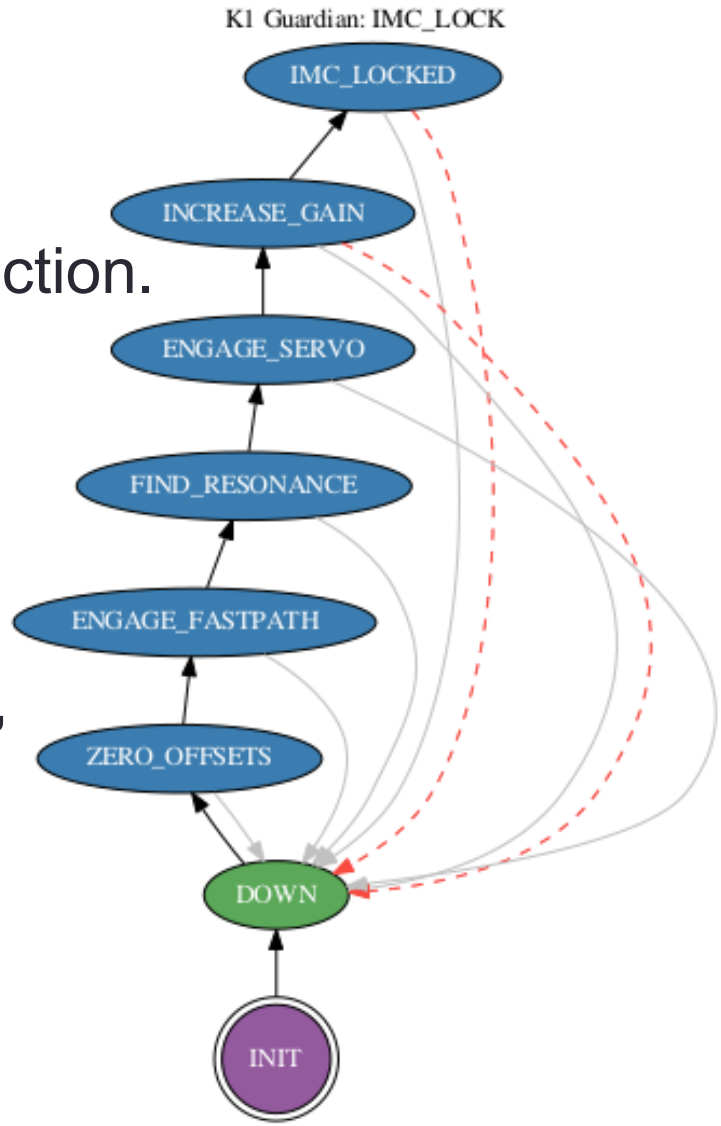
IMC-MCL_SERVO_OFFSET

IMC-MCL_SERVO_GAIN

Automation IMC LOCK with Guardian

- Guardian for IMC LOCK consist of many state.
- Each state mainly consist of two function.
  - ✓main function
  - ✓run function
- When function return true, state move to next state.
- If main or run function return DOWN, then jump to DOWN state.

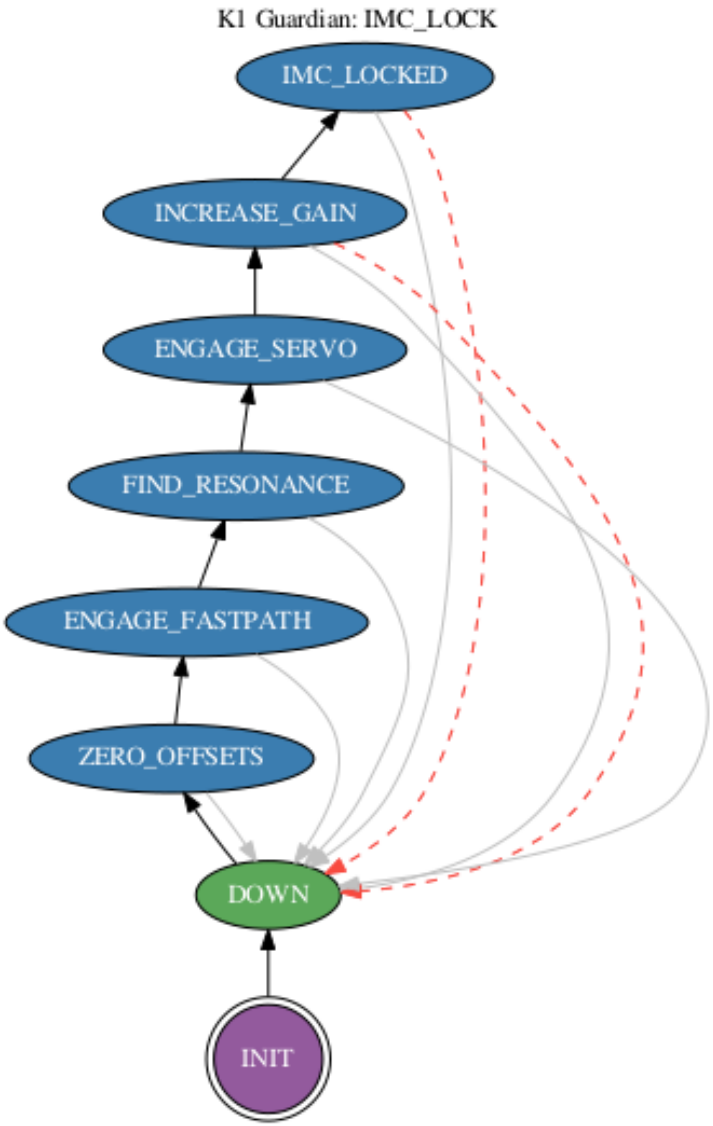




Automation IMC LOCK with Guardian

- Description of each state
- DOWN :
  - reset system
  - after lock loss
- ZERO_OFFSETS:
  - zero any offset
  - before engaging servo
- FIND_RESONANCE
  - move mirror to find resonance condition
  - before engaging slow servo
- IMC_LOCKED:
  - watch for lockloss

  →NOMINAL_STATE (disired)



Guardian graph in python

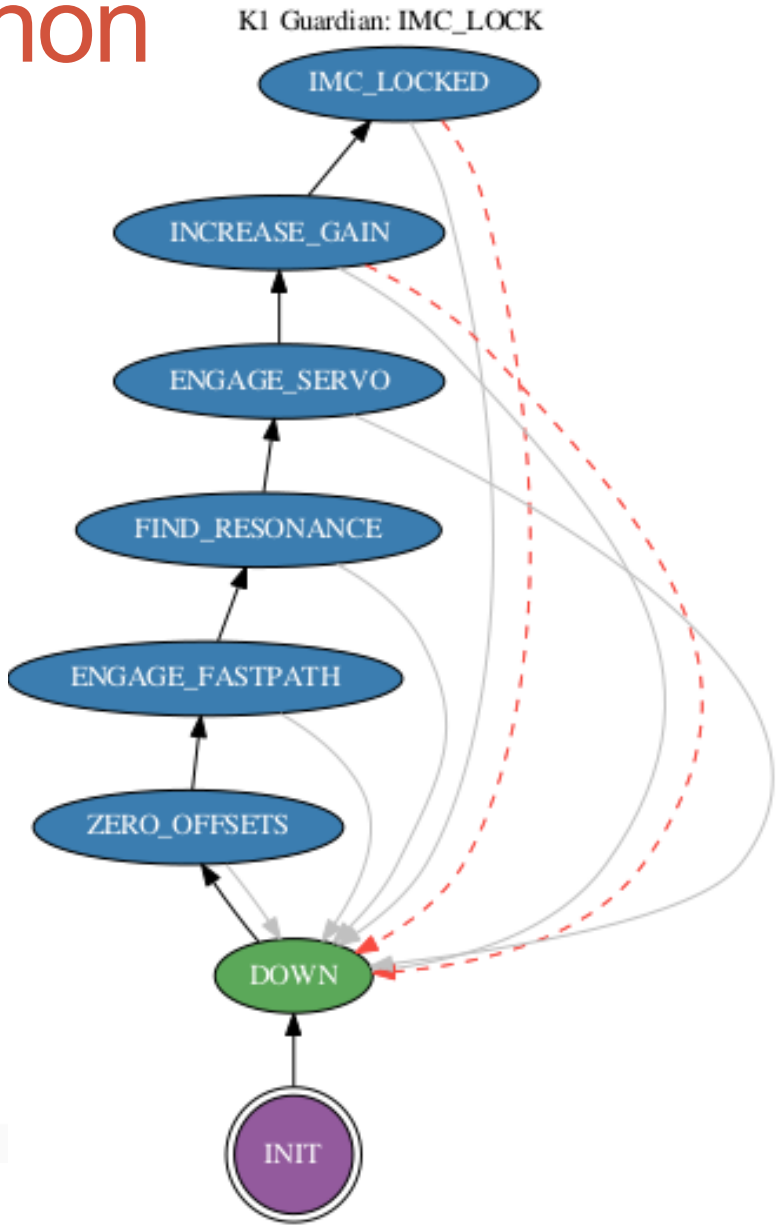
```
# STATE definitions

class INIT(GuardState):
    index = 0


class DOWN(GuardState):
    index = 1
    goto = True
    def main(self):
        #ezca.switch('IMC-MCL_SERVO', 'OUTPUT', 'OFF')
        ezca['IMC-MCL_SERVO_GAIN'] = 0
        ezca['IMC-MCL_SERVO_OFFSET'] = 0


class ZERO_OFFSETS(GuardState):
    index = 10
    def main(self):
        ezca['IMC-MCL_SERVO_OFFSET'] = 0
        pass

class ENGAGE_FASTPATH(GuardState):
    index = 15
    def main(self):
        ezca['IMC-REFL_SERVO_FASTEN'] = 'ON'
        pass
```



Guardian graph in python

```
class FIND_RESONANCE(GuardState):
    index = 20
    def main(self):
        self.step = 1.0
        self.tramp = 0.005

        self.step *= -math.copysign(1, ezca['VIS-MCE_TM_TEST_L_OFFSET'])
        ezca['VIS-MCE_TM_TEST_L_TRAMP'] = self.tramp

        log("looking for resonance: step=%s, tramp=%s" % (self.step, self.tramp)
)

    def run(self):
        if is_imc_locked():
            log('resonance found.  TRANS = %s' % ezca['IMC-MCL_TRANS_OUTPUT'])
            return True
        else:
            ezca['VIS-MCE_TM_TEST_L_OFFSET'] += self.step
            time.sleep(self.tramp)

# utility functions

def is_imc_locked():
    """return True if IMC is locked (TRANS above threshold)"""
    return ezca['IMC-MCL_TRANS_OUTPUT'] > 0.04
```

Guardian graph in python

```
class ENGAGE_SERVO(GuardState):
    index = 30

    def main(self):
        #ezca.switch('IMC-MCL_SERVO', 'OUTPUT', 'ON')
        ezca['IMC-MCL_SERVO_GAIN'] = -0.72


class INCREASE_GAIN(GuardState):
    index = 40

    @imc_lock_check
    def main(self):
        #ezca['IMC-MCL_SERVO_GAIN'] = 1
        pass


class IMC_LOCKED(GuardState):
    index = 100

    @imc_lock_check
    def run(self):
        return True
```

```
# STATE decorators

class imc_lock_check(GuardStateDecorator):
    """Decorator to check that IMC is locked"""
    def pre_exec(self):
        if not is_imc_locked():
            return 'DOWN'
```