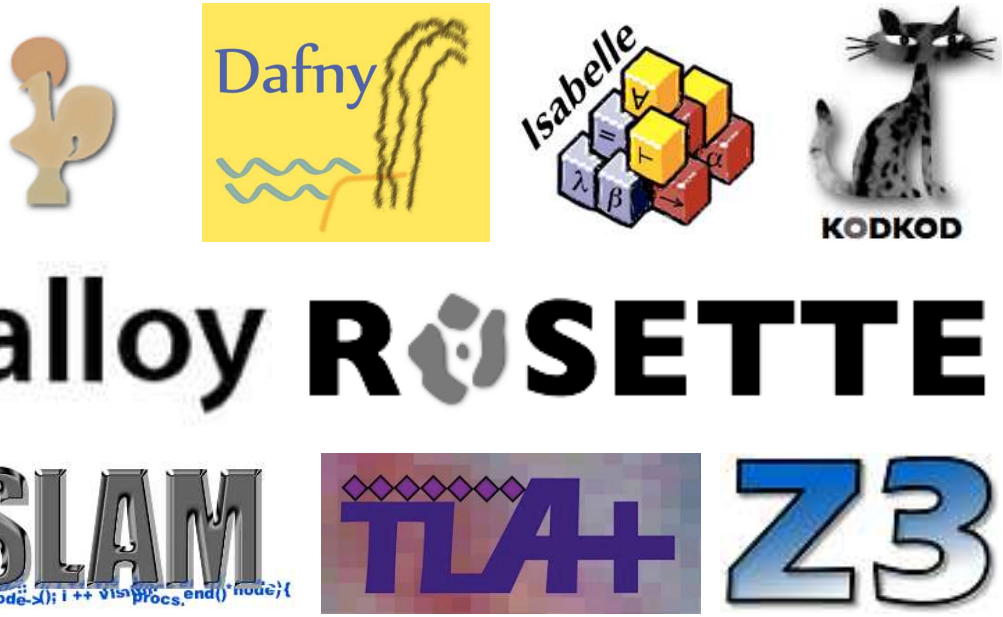
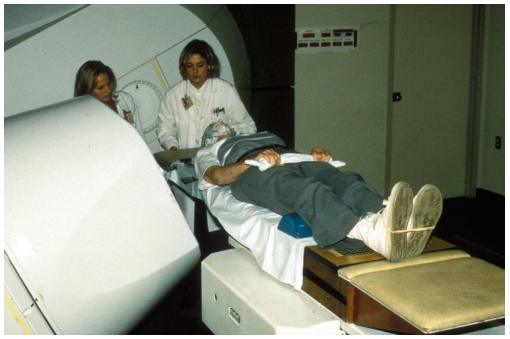
Investigating Safety of a Radiotherapy Machine Using System Models with Pluggable Checkers

Combining Formal Models with Concrete Evidence

Stuart Pernsteiner, Calvin Loncaric, Emina Torlak, Zachary Tatlock, Xi Wang, Michael D. Ernst, and Jonathan Jacky

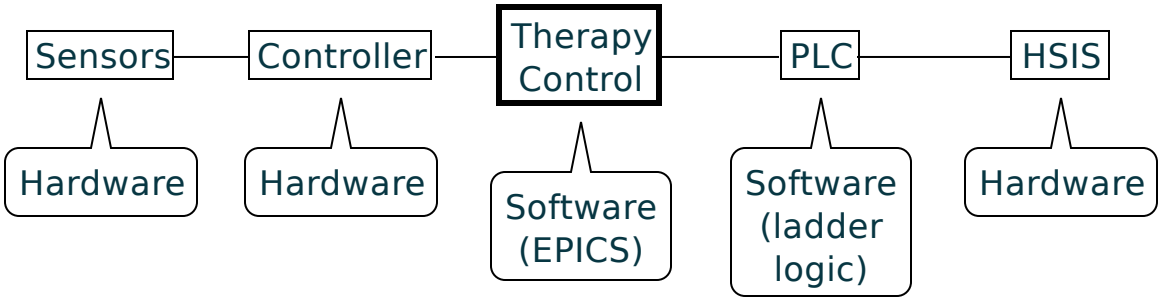

Dafny
Isabelle
KODKOD
alloy
ROSETTE
SLAM
TLA+
Z3

CNTS Safety Property



Prescription Safety:
If any setting exceeds the prescribed tolerances
then the beam will shut off.

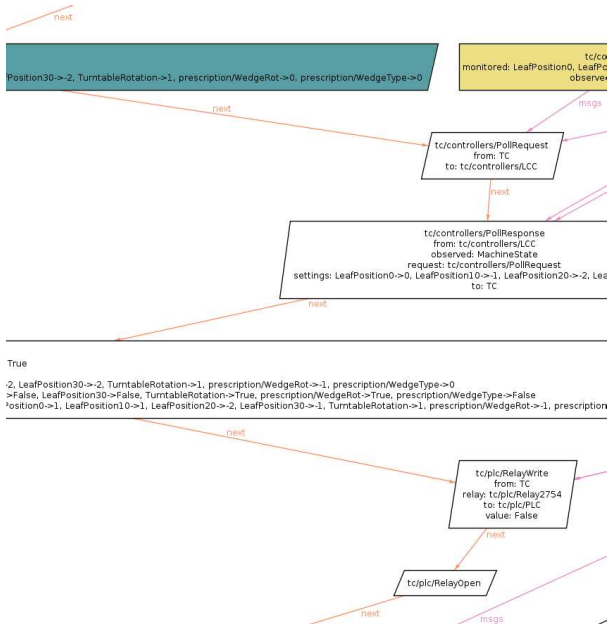
CNTS Architecture



Prescription Safety:
If the gantry angle exceeds the prescribed tolerances then the beam will shut off.

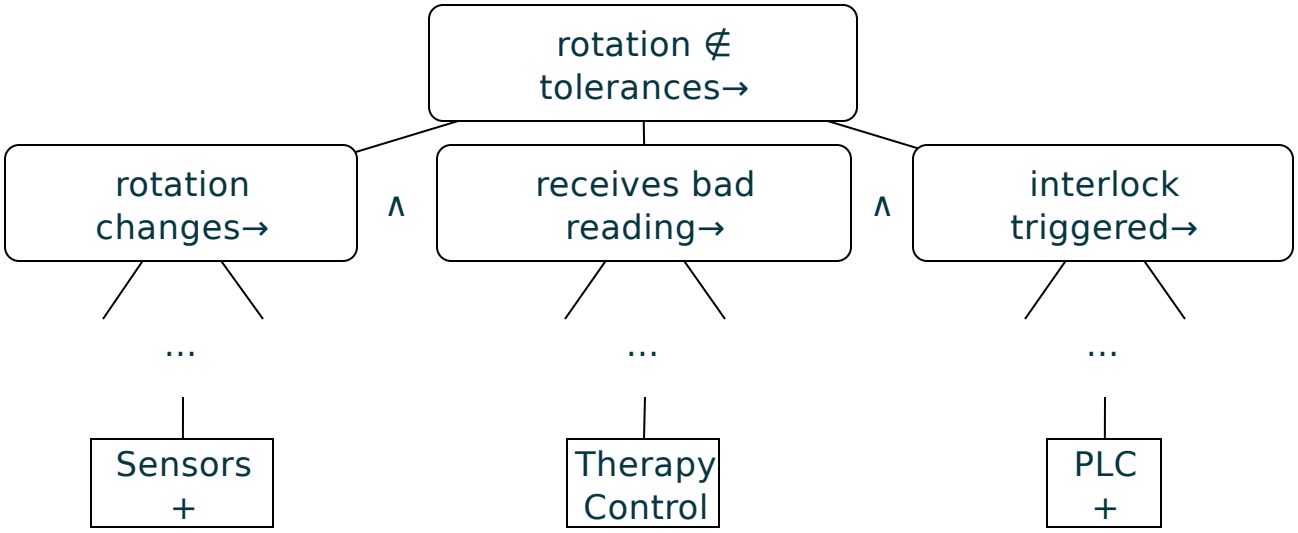
Formal Modeling

```
pred ControllersAreCorrect[] {
    evidence[Expert,
             "--file" -> "cnts-sc/rx/inspections.yaml" +
             "--claim" -> "assume-controllers-ok",
             ManualInspection] =>
    {
        ((all c: Controller | c.observed = MachineState) and
         (all req : PollRequest |
            (let resp = request.req |
                one resp and
                resp in req.next and
                resp.request = req and
                resp.to = req.from and
                resp.from = req.to and
                resp.settings = (resp.from.monitored)<:(resp.observed.actual) and
                resp.observed in resp.from.observed and
                happensMostRecentlyBefore[resp.observed, req, resp.from.observed])))
    }
}

-- TC correctly updates MOD1:Waveform:Calc for every PollResponse it receives
-- from the given embedded controller.
pred TCProcessesPollResponsesCorrectly[controller : Controller] {
    all resp: TC.receivedMsgs & PollResponse & from.controller |
        one tcdb :  TC.db & resp.next |
            -- All variables corresponding to observed machine settings are set correctly.
            resp.settings in tcdb.actual and

            -- The therapy sum interlock bit is 0 if the actual setting value is outside of the
            -- tolerance and the override flag is not raised for that setting.
            ((some setting :  controller.monitored |
              tcdb.actual[setting] not in tcdb.tolerated[setting] and
              tcdb.overridden[setting] = False) =>
             tcdb.MOD1_WaveForm_Calc_1 = False) and

            -- The waveform output record gets processed as part of the update.
            tcdb.MOD1_IntlkCnOutWArray_Processed = True
}

-- If PLC relay 2754 is opened, then coil 1623 is immediately deenergized,
-- which is immediately conveyed to HSIS through a CoilChange message.
pred Coil1623DeenergizedWhenRelay2754Open [] {
    evidence[PLC_Analysis,
             "--mode" -> "all-paths-to-coil-contain-relay" +
             "--network-file" -> "plc-code/cyclotron/mod1.stu" +
             "--coil" -> "%M1623" +
             "--relay" -> "%M2754",
             Proof] =>
    {
        all relayOpen: Relay2754.state & RelayOpen |
            some coilState: Coil1623.state & CoilDeenergized, coilChange : PLC.sentMsgs & CoilChange |
                coilState in relayOpen.next and
                coilChange in coilState.next and
                coilChange.coil = Coil1623 and
                coilChange.state = coilState
    }
}
```



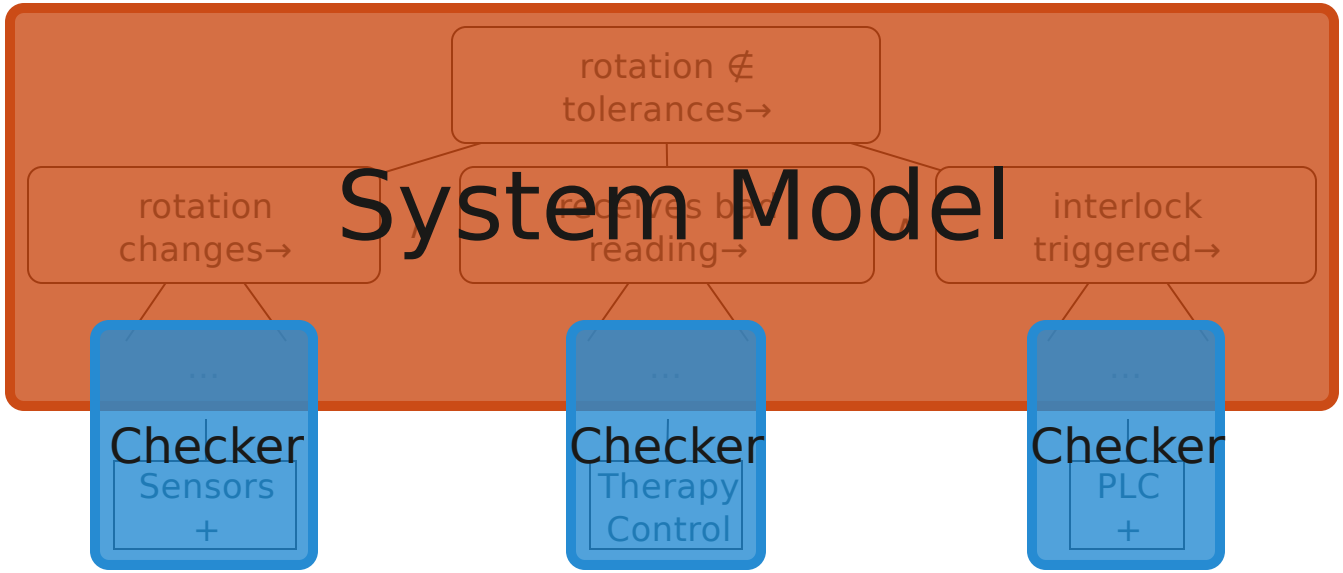
+ Automatically check safety of the model
− Difficult to ensure the model matches the system

Safety Property Decomposition



\+ Easy to integrate any type of evidence
− No support for automated checking

Our Approach



+ Automated checking of safety properties
+ Can incorporate any type of evidence
★ Bonus: Building the model eases checker development

Outline

- Background
-  Modelling with Evidence
-  Pluggable Checker Development
- Results

The Alloy Model

Given:

- When the sensor reading changes,
  the controller outputs the new reading.
- When the therapy control program processes an event,
  if the current rotation reading is out of tolerances,
  it sets the Therapy Sum Interlock value to 0.
- When PLC relay #2754 is opened,
  PLC coil #1623 is deenergized.
- ...

Is it the case that:

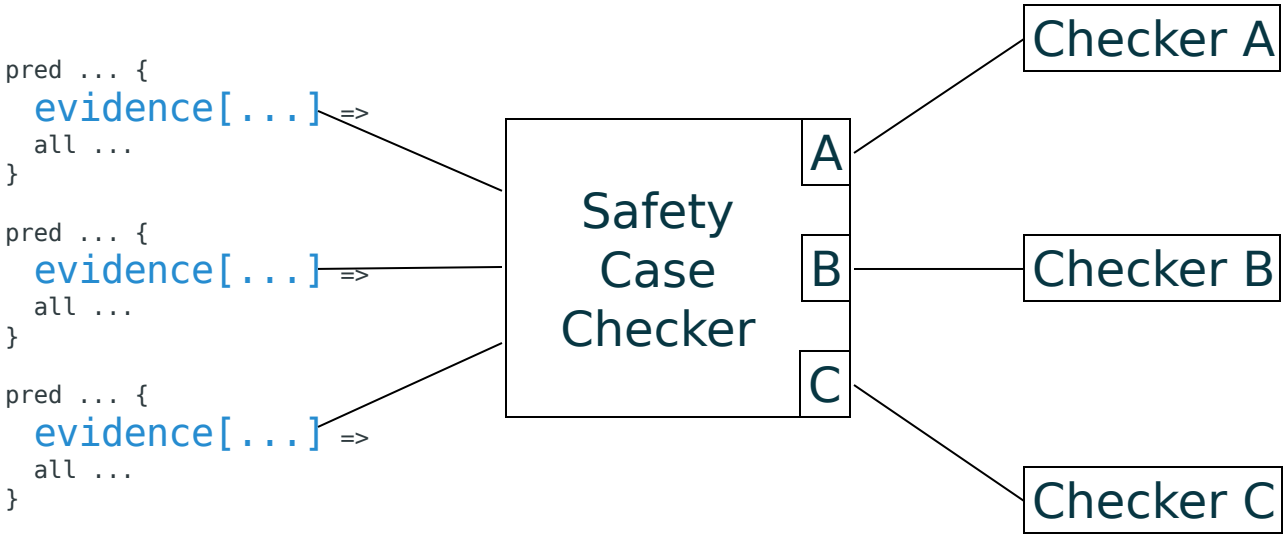
- When the gantry rotation angle moves out of the prescribed tolerances, the beam shuts off.

```
pred ControllersAreCorrect[] {
    evidence[Expert,
             "--file" -> "cnts-sc/rx/inspections.yaml" +
             "--claim" -> "assume-controllers-ok",
             ManualInspection] =>
    {
        ((all c: Controller | c.observed = MachineState) and
         (all req : PollRequest |
            (let resp = request.req |
               one resp and
               resp in req.next and
               resp.request = req and
               resp.to = req.from and
               resp.from = req.to and
               resp.settings = (resp.from.monitored)<:(resp.observed.actual) and
               resp.observed in resp.from.observed and
               happensMostRecentlyBefore[resp.observed, req, resp.from.observed])))
    }
}

-- TC correctly updates MOD1:Waveform:Calc for every PollResponse it receives
-- from the given embedded controller.
pred TCProcessesPollResponsesCorrectly[controller : Controller] {
    all resp: TC.receivedMsgs & PollResponse & from.controller |
        one tcdb :  TC.db & resp.next |
            -- All variables corresponding to observed machine settings are set correctly.
            resp.settings in tcdb.actual and

            -- The therapy sum interlock bit is 0 if the actual setting value is outside of the
            -- tolerance and the override flag is not raised for that setting.
            ((some setting :  controller.monitored |
              tcdb.actual[setting] not in tcdb.tolerated[setting] and
              tcdb.overridden[setting] = False) =>
             tcdb.MOD1_WaveForm_Calc_1 = False) and

            -- The waveform output record gets processed as part of the update.
            tcdb.MOD1_IntlkCnOutWArray_Processed = True
}

-- If PLC relay 2754 is opened, then coil 1623 is immediately deenergized,
-- which is immediately conveyed to HSIS through a CoilChange message.
pred Coil1623DeenergizedWhenRelay2754Open [] {
    evidence[PLC_Analysis,
             "--mode" -> "all-paths-to-coil-contain-relay" +
             "--network-file" -> "plc-code/cyclotron/mod1.stu" +
             "--coil" -> "%M1623" +
             "--relay" -> "%M2754",
             Proof] =>
    {
        all relayOpen: Relay2754.state & RelayOpen |
            some coilState: Coil1623.state & CoilDeenergized, coilChange : PLC.sentMsgs & CoilChange |
                coilState in relayOpen.next and
                coilChange in coilState.next and
                coilChange.coil = Coil1623 and
                coilChange.state = coilState
    }
}

run BeamShutsOffDueToOOTSetting {
        some ms : MachineState |
                system and
                ms.actual not in Prescription.tolerated and
                not badSettingOverriden[ms] and
                (some on : Beam.state & BeamOn | happensBefore[on, ms]) and
                (some off : Beam.state & BeamOff | happensBefore[ms, off]) and
                (all tcdb: TCdb | happensBefore[BeamOn, tcdb] and happensBefore[tcdb, BeamOff]) and
                (all tcdb : TCdb | lone tcdb.~next & PollResponse)
} for 3 but 10 Event,  2 int
```

Integrating Evidence

```
pred TCRotationCheckCorrect [] {
  evidence[EPICS_SE, "--prop" -> "tc_intlk"] =>
  all reading: TC.receivedMsgs |
    reading.value not in Rx.tolerated =>
    some interlock: TC.sentMsgs & reading.next
      interlock.ok = false
}
```

Integrating Evidence



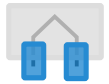
Outline

- Background
-  Modelling with Evidence
-  Pluggable Checker Development
- Results

CNTS Checkers

- EPICS linter
- EPICS verifier
- PLC checker
- EPICS-PLC connection checker
- Expert assertion checker

EPICS Verifier



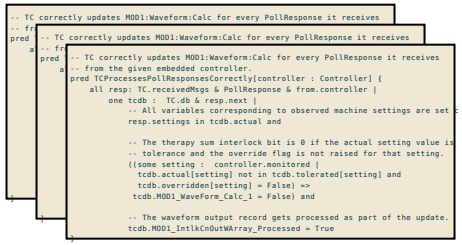
Starting from an arbitrary program state,
when the therapy control program processes an event,
if the current rotation reading is out of tolerances,
it sets the Therapy Sum Interlock value to 0.

```
(define (tc_intlk)
  (process_IsoGantryActual)

  ; ...

  (assert (=>
    (> (abs (- prescribed actual)) tolerance)
    (= beam-interlock 0))))
```

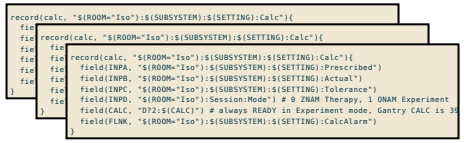
EPICS Verifier

- Input/output
- Field links
- Dynamic link modification
- Calculation
- Array calculation
- String manipulation
- Process-on-init
- Timer callbacks
- ...

Properties:



Implementation:



Not all language features are relevant to the properties!

EPICS-PLC Connection Checker

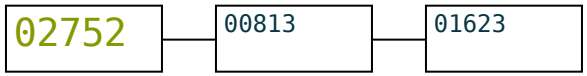
mod1_intlk_outputs.db:

```
record(waveform, MOD1:IntlkCnOutWArray) {
    record(DTYP, "asynInt32ArrayOut")
    field(INP, "@asyn(therapyIntlkOut_Word 0)MODBUS_DATA")
    field(FTVL, "LONG")
    field(NELM, "6")
}
```

st.cmd:

```
drvModbusAsynConfigure("therapyIntlkOut_Word",
    "therapyIntlkWrite", 9, 15, 2752, 6, 0, 1, "Modicon")
```

mod1.stu:



Outline

- Background
-  Modelling with Evidence
-  Pluggable Checker Development
- Results

Results

We found real bugs:

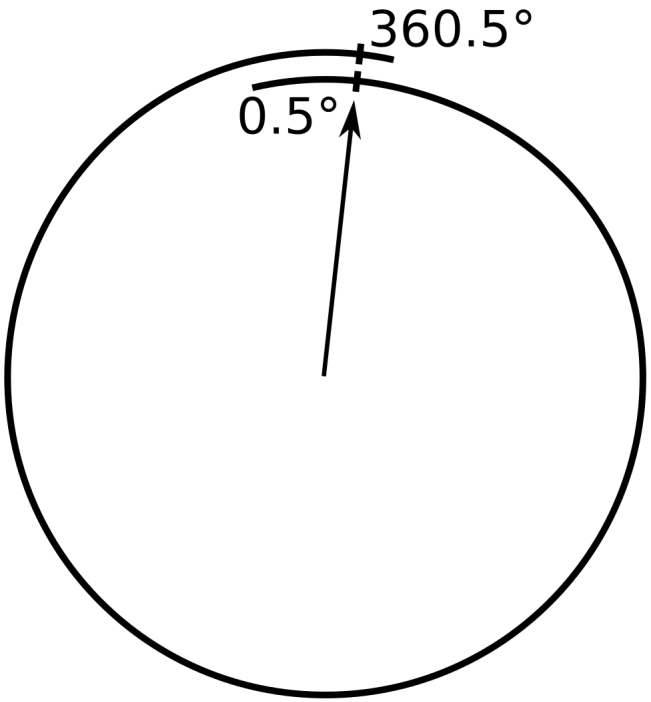
Bad gantry rotation check
- Arithmetic error; *beam may fail to shut off*

Array indexing discrepancy
- Off-by-one error; *beam may fail to shut off*

Broken dataflow links
- System reads undefined values; *errors may not be reported*

Missing PLC relay
- Initial system model did not correspond to reality

Gantry Rotation Bug



Gantry Rotation Bug

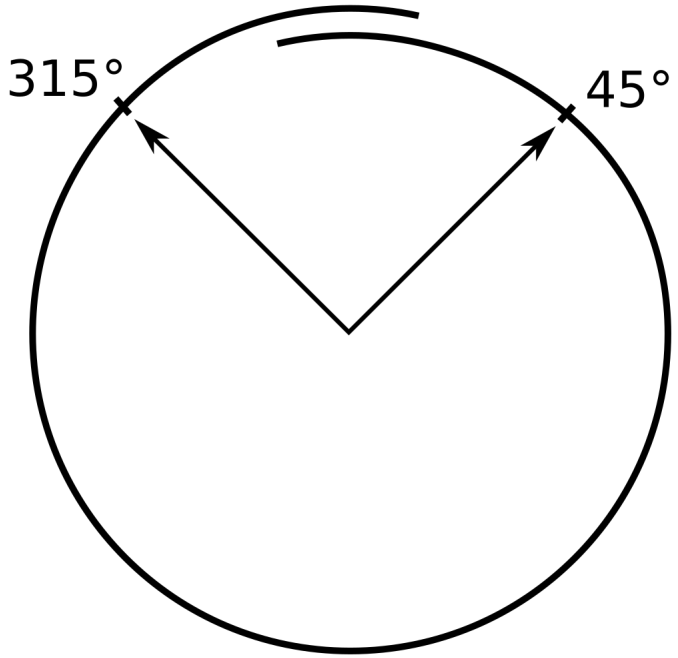
"We found a bug..."

- Something is wrong with the handling of gantry rotation
- The safety property can be violated

EPICS Verifier counterexample:

```
(("Gantry:Rotation:Prescribed" 315)
 ("Gantry:Rotation:Actual" 45)
 ...)
```

Gantry Rotation Bug



Gantry Rotation Bug

"We found a bug..."

- Something is wrong with the handling of gantry rotation
- The safety property can be violated

EPICS Verifier counterexample:

```
(("Gantry:Rotation:Prescribed" 315)
 ("Gantry:Rotation:Actual" 45)
 ...)
```

"There appears to be an error on line 29 of `gantry_couch.substitutions`..."

- We verified a real CNTS safety property using a system model and pluggable checkers
- Modeling helped guide and simplify checker development
- We found real bugs in a safety-critical system