Model Predictive Control of Building Dynamics

Edward J. Xu

2019-09-15

Initialization

```
library(tidynamics)
library(tibble)
library(tidyverse)
library(gridExtra)
library(egg)
#> Warning: package 'egg' was built under R version 3.5.2

source("~/GitHub/tidynamics/vignettes/funcs/mpc/asHours.R")
source("~/GitHub/tidynamics/vignettes/funcs/mpc/asP.R")
source("~/GitHub/tidynamics/vignettes/funcs/mpc/prbs.R")
source("~/GitHub/tidynamics/vignettes/funcs/mpc/sim_building.R")
#> Warning: package 'lpSolve' was built under R version 3.5.2
#> Warning: package 'expm' was built under R version 3.5.2
```

If you have a working GLPK solver on your system and want to implement a computationally more optimized controller use the following

- install.packages("curl")
- install.packages("devtools")
- library(devtools)
- install_version("slam",version="0.1-40", repos = "http://cran.us.r-project.org")

Now install the Rglpk which is used for solving linear programs

- install.packages("Rglpk")
- library(Rglpk)

1, Get Data

```
ti <-
  read_csv(
    "~/GitHub/tidynamics/data/mpc.csv",
    skip = 1,
    col_names = c("timedate", "Y1", "Y2", "Ta", "Gv", "Ph1", "Ph2")
  ) %>%
  mutate(timedate = asP(.$timedate)) %>%
  mutate(t = asHours(.$timedate - .$timedate[1]))
```

```
ti %>%
  print()
#> # A tibble: 540 x 8
#>    timedate               Y1    Y2    Ta       Gv   Ph1   Ph2     t
#>    <dttm>              <dbl> <dbl> <dbl>    <dbl> <dbl> <dbl> <dbl>
#>  1 1983-10-10 15:20:00  24.5  27.4  11.8 0.007      1.5   1.5 0
#>  2 1983-10-10 15:30:00  24.6  27.4  11.9 0.0053     1.5   1.5 0.167
#>  3 1983-10-10 15:40:00  24.7  27.3  12.1 0.0049     1.5   0   0.333
#>  4 1983-10-10 15:50:00  24.8  27.2  12.3 0.0051     1.5   0   0.5
#>  5 1983-10-10 16:00:00  24.7  27.0  12.6 0.0011     1.5   1.5 0.667
#>  6 1983-10-10 16:10:00  24.6  26.9  12.7 0.001      1.5   0   0.833
#>  7 1983-10-10 16:20:00  24.8  26.8  12.7 0.0007     1.5   0   1
#>  8 1983-10-10 16:30:00  24.8  26.7  12.7 0.0001     1.5   0   1.17
#>  9 1983-10-10 16:40:00  24.9  26.9  12.7 0          1.5   1.5 1.33
#> 10 1983-10-10 16:50:00  24.9  27.2  12.7 0.000600   1.5   1.5 1.5
#> # … with 530 more rows
```

2, Model using CTSM

```
ti_est <- tibble(
  name = c(
    "T1a0", "T1m0", "C1a", "C1m", "R1a",
    "R1m", "A1w", "p1", "p1a", "p1m", "e11"
  ),
  init = c(25, 25, 6, 12, 10, 1, 1, 0.5, 1, 1, -1),
  lb = c(0, 0, 1E-5, 1, 1, 1E-10, 1E-10, 0, -30, -30, -50),
  up = c(35, 35, 20, 50, 80, 10, 10, 1, 10, 10, 10)
)

li_mod <- list()

li_mod[[1]] <- set_mod_ctsm(
  c_expr_sys = c(
    d(T1a) ~ (
        1 / (C1a * R1m) * (T1m - T1a) + 1 / (C1a * R1a) * (Ta - T1a) +
        1 / C1a * Ph1 + p1 * A1w / C1a * Gv
      ) * d(t) + exp(p1a) / C1a * d(w1a),
    d(T1m) ~ (
        1 / (C1m * R1m) * (T1a - T1m) + (1 - p1) * A1w / C1m * Gv
      ) * d(t) + exp(p1m) / C1m * d(w1m)
    ),
  expr_obs = Y1 ~ T1a,
  expr_var = Y1 ~ exp(e11),
  c_input = c("Ta", "Ph1", "Gv"),
  ti_est = ti_est
)
```

3, Estimate the Model

4, Get State Space Model

```
li_mat_ss <-
  fit1 %>%
  trans_ctsm_ss()

li_mat_ss_d <-
  li_mat_ss %>%
  trans_mat_ss(ti_data = ti)
```

5, Simulate the Control Process

```
result <- sim_building(li_mat_ss, li_mat_ss_d, ti)
Tall <- result$Tall
Ymin <- result$Ymin
Ymax <- result$Ymax
Price <- result$Price
Tmax <- result$Tmax
u <- result$u

ti_result <- tibble(
  t = 1:length(u),
  tall1 = Tall[1,],
  tall2 = Tall[2,],
  u = u
  )
ti_p <- tibble(
  t = 1:length(result$Price),
  p = result$Price
)
```