# CobotOps Analysis Data Dictionary

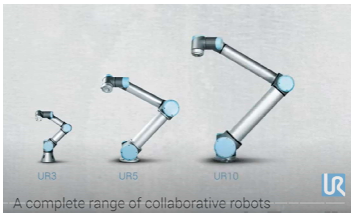
**some information is assumed and should be revised in receipt of accurate domain knowledge*

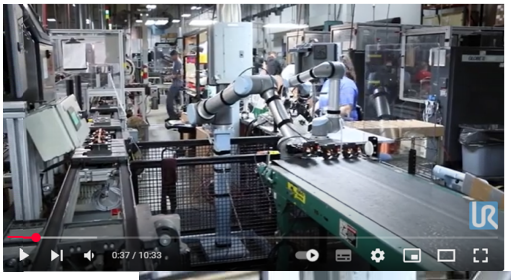
## *Dataset and UR3 Cobot(collaborative robot) background*

M. Tyrovolas, K. Aliev, D. Antonelli, and C. Stylios. "UR3 CobotOps," UCI Machine Learning Repository, 2024. [Online]. Available: https://doi.org/10.24432/C5J891.

The UR3 CobotOps Dataset is an essential collection of multi-dimensional time-series data from the UR3 cobot, offering insights into operational parameters and faults for machine learning in robotics and automation. It features electrical currents, Currents, speeds across joints (J0-J5), gripper current, operation cycle count, protective stops, and grip losses, collected via MODBUS and RTDE protocols. This dataset supports research in fault detection, predictive maintenance, and operational optimization, providing a detailed operational snapshot of a leading cobot model for industrial applications

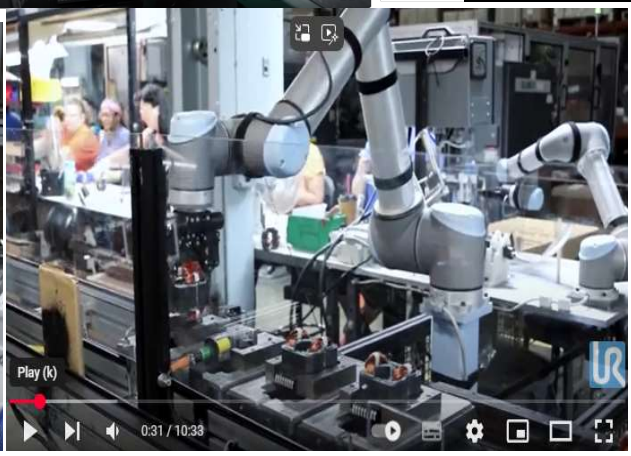
I. all features continuous except cycle, grip lost and protective stop




• Dataset Reference: M. Tyrovolas, K. Aliev, D. Antonelli, and C. Stylios. "UR3 CobotOps," UCI Machine Learning Repository, 2024. https://doi.org/10.24432/C5J891

• Video reference: Robotics applications by Universal Robots - Easy Automation with Collaborative Robots [ 2020 ]
• Collaboration at all levels – experience Universal Robots in 21 applications at Hanover Fair 2017


- *Video references:*
  - *Robotics applications by Universal Robots - Easy Automation with Collaborative Robots [ 2020 ]*
  - *Collaboration at all levels – experience Universal Robots in 21 applications at Hanover Fair 2017*

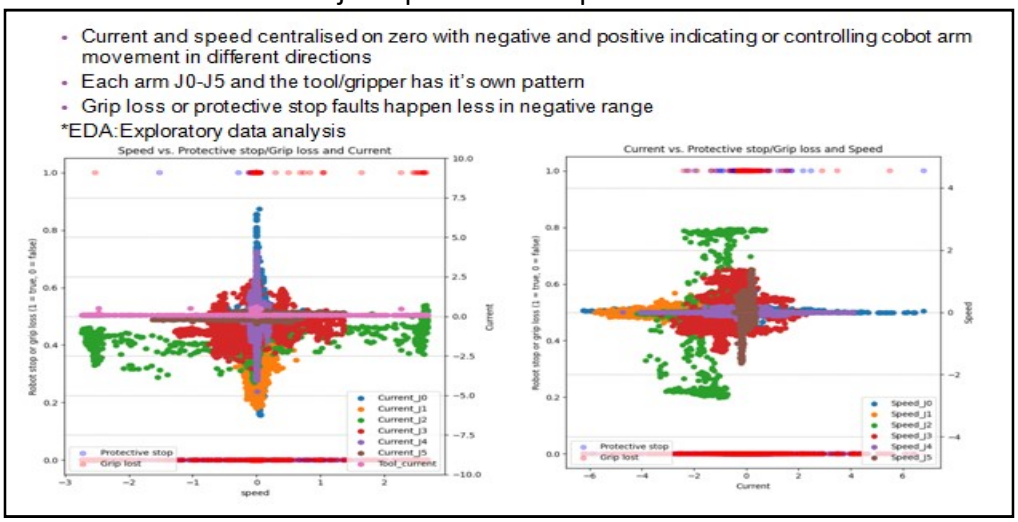
*Data Dictionary (1. Orig dataset, 2. Columns/features added to Orig, 3. New summary table)*

1. Original dataset

```
## view some statistical details
dataset.describe().T
```

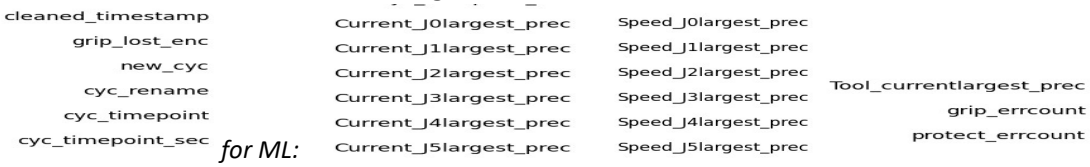
| | count | mean | std | min | 25% | 50% | 75% | max |
|---|---|---|---|---|---|---|---|---|
| Num | 7409.0 | 3705.000000 | 2138.938405 | 1.000000 | 1853.000000 | 3705.000000 | 5557.000000 | 7409.000000 |
| Current_J0 | 7363.0 | -0.048934 | 0.829395 | -6.247756 | -0.195026 | -0.080830 | 0.095959 | 6.806938 |
| Temperature_T0 | 7355.0 | 34.906781 | 2.762057 | 27.812500 | 33.000000 | 36.500000 | 37.062500 | 37.250000 |
| Current_J1 | 7355.0 | -2.297936 | 0.800450 | -5.808734 | -2.664919 | -2.236342 | -1.778334 | 1.083596 |
| Temperature_J1 | 7355.0 | 37.659636 | 3.247315 | 29.312500 | 35.375000 | 39.687500 | 40.125000 | 40.500000 |
| Current_J2 | 7355.0 | -1.199381 | 0.609984 | -4.171966 | -1.552803 | -1.077137 | -0.838721 | 2.464940 |
| Temperature_J2 | 7355.0 | 38.064064 | 3.311948 | 29.375000 | 35.750000 | 40.187500 | 40.437500 | 40.937500 |
| Current_J3 | 7355.0 | -0.605312 | 0.514937 | -3.333102 | -0.830933 | -0.571190 | -0.388398 | 2.270268 |
| Temperature_J3 | 7355.0 | 40.936999 | 3.182399 | 32.125000 | 38.937500 | 43.062500 | 43.125000 | 43.437500 |
| Current_J4 | 7355.0 | -0.022968 | 0.630789 | -4.738406 | -0.125809 | -0.012325 | 0.086098 | 4.089389 |
| Temperature_J4 | 7355.0 | 42.605167 | 3.677670 | 32.250000 | 40.375000 | 45.062500 | 45.187500 | 45.375000 |
| Current_J5 | 7355.0 | 0.000904 | 0.129141 | -0.474556 | -0.094875 | 0.014496 | 0.077995 | 0.392547 |
| Temperature_J5 | 7355.0 | 41.891094 | 3.680003 | 32.000000 | 39.500000 | 44.375000 | 44.500000 | 44.937500 |
| Speed_J0 | 7355.0 | 0.001671 | 0.148826 | -0.656331 | -0.000483 | 0.000000 | 0.002115 | 0.791929 |
| Speed_J1 | 7355.0 | 0.000061 | 0.049370 | -0.330833 | -0.000659 | 0.000000 | 0.000540 | 0.615526 |
| Speed_J2 | 7355.0 | 0.003791 | 0.729422 | -2.733091 | -0.004100 | 0.000000 | 0.000205 | 2.679848 |
| Speed_J3 | 7355.0 | -0.004610 | 0.359999 | -1.271460 | -0.005365 | 0.000000 | 0.009912 | 1.363093 |
| Speed_J4 | 7355.0 | -0.000810 | 0.038436 | -0.226233 | -0.000120 | 0.000000 | 0.000066 | 0.193914 |
| Speed_J5 | 7355.0 | 0.002880 | 0.313953 | -1.629721 | -0.000803 | 0.000000 | 0.012924 | 1.378015 |
| Tool_current | 7355.0 | 0.109527 | 0.078832 | 0.020180 | 0.082387 | 0.085192 | 0.089518 | 0.602054 |
| cycle | 7409.0 | 141.113241 | 80.542340 | 1.000000 | 68.000000 | 154.000000 | 218.000000 | 264.000000 |
| Robot_ProtectiveStop | 7355.0 | 0.037797 | 0.190719 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 1.000000 |

   a. Num – unique number for each data row, similar to an index
   b. Timestamp – format 2022-10-26T08:17:22.852Z, requires extraction
   c. 6 moving joints on UR3 cobot allowing large flexibility of movement – J0 to J5
      i. Temperature – temperature measured at each joint per timestamp
         - Temperature_T0, likely should be Temperature_J0, unchanged
      ii. Current – Electrical current driving robot actions, with positive and negative values, being measured at each joint per timestamp
      iii. Speed – speed of each joint, with positive and negative values, being measured at each joint per timestamp




   d. Tool_current – current driving the gripper/tool that picks up/drops items as the arm moves, with positive and negative values, measured per timestamp
   e. Cycle – self contained programmed cycle of actions for robot to perform
   f. Robot_Protective_stop – robot stops pre-emptively in response to sensor signals to avoid an issue, value 0 for false or 1 for true if occurs at timestamp
   g. Grip_lost – tool/gripper loses or detects lost grip when programming tells it that it should have item gripped, false or true Boolean values per timestamp

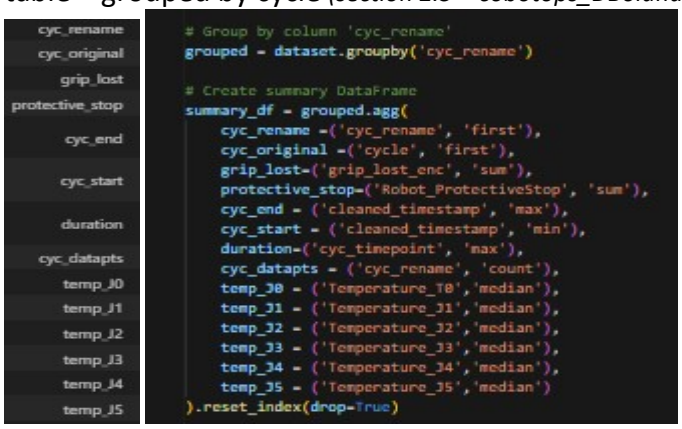
2. Columns/features added to Original




*for ML:*

   a. **Understanding cycle behaviour** *(section 2.2 notebook cobotops_DBolandMar25.ipynb)*
      i. Cleaned_timestamp – extracted to format for calculations with datetime 2022-10-26 08:17:21.847000. datatype = datetime
      ii. Date – date only
      iii. Time – time only
      iv. Grip_lost_enc – Boolean variable converted false to 0 and true to 1
      v. Time_str = time saved as a string for plotting initial graphs (unable to do scatter this way so later used graphs with cleaned_timestamp)
      vi. New_cyc = order of cycles run was not sequential and some were repeated. Created boolean variable that returns true if previous row cycle value same as current row and false if not
      vii. Cyc_rename = if new_cyc is false i.e. not the same as previous row, start a new cycle number. Generates sequential cycle number in order of action taken. Comparable to timestamp as lowest earliest cycle and highest last cycle (could have been combined with previous to prevent having another column).
      viii. Cyc_timepoint = variable to determine what point of cycle a datapoint is. Could be interesting to compare at similar timepoints, of different cycles, what robot was doing. Datatype = timedelta
      ix. Cyc_timepoint_sec = convert above variable to float of seconds, to be used for plotting
   b. **Machine learning features** – add new variable from a previous timepoint (lagging data) to each unique datapoint so can model data with traditional classification models but incorporating time series element. *(section 3.3.1 notebook cobotops_DBolandMar25.ipynb)*
      i. For current and speed look back at previous 10 rows/timepoints and take the value of highest magnitude (positive or negative)
         - 12 features named: Current/Speed_J0..5largest_prec
         - Also for tool/gripper: Tool_currentlargest_prec
      ii. For errors/faults count up the number of datapoints where there hasn't been an error i.e. if no fault on previous row increase the counter but if fault occurs on previous row set to -1
         - 2 features: grip_errcount and protect_errcount

3. New summary table – grouped by cycle *(section 2.5 - cobotops_DBolandMar25.ipynb)*




   a. Cyc_rename = grouping variable, sequential cycle naming to how they ran on robot, comparable to timestamp as lowest earliest and highest last cycle
   b. Cyc_original = original cycle naming, order run was not sequential and some were repeated - maintain as is recorded cycle and may have other meaning
   c. Grip_lost = sum of all grip loss errors in the cycle
   d. Protective_stop = sum of all protective stop errors in the cycle
   e. Cyc_end = max timestamp in the cycle, end of cycle
   f. Cyc_start = min timestamp in the cycle, start of cycle
   g. Duration = max cycle_timepoint, i.e. duration of cycle. Datatype = timedelta
   h. Cyc_datapoints = count of number of datapoints in a cycle
   i. Temp_J0 . . J5 = median temperature per cycle for each robot joint (J0-J5)
   j. All_errors = sum of grip_lost and protective_stop, sum all errors in the cycle
   k. Break_post_cyc = calculate length of break between cycles at the end/post the current cycle(s). Subtraction of cyc_end from cyc_start of next cycle/next row in dataframe (cyc_start). Datatype = timedelta.
   l. Duration_sec = convert timedelta duration to float (secs) for plotting
   m. Break_post_c_sec = convert timedelta break_post_cyc to float (secs) for plotting