Robotic Arm Dataset (RAD) Description

1 Command Dataset

The centrifuge is the simplest of all modules. It is connected via the C9 controller, and can be directed to start and stop spinning using the `OUTP n` command exposed by the C9 controller, where n denotes the port to which the centrifuge is connected (table 1).

Command	Arguments (type) / Responses (type)	Description
OUTP n	Status (b) / -	Directs centrifuge to start and stop spinning

Table 1: Fisherbrand 100-240V 50/60Hz Mini-Centrifuge

Command	Arguments (type) / Responses (type)	Description
START_1	- / -	Start the heater
STOP_1	- / -	Stop the heater
START_4	- / -	Start the motor
STOP_4	- / -	Stop the motor
SET_MODE_A	- / -	Set operating mode A
SET_MODE_B	- / -	Set operating mode B
SET_MODE_D	- / -	Set operating mode D
RESET	- / -	Switch to normal operating mode
OUT_SP_1	Temperature (f) / -	Adjust the set temperature value
OUT_SP_4	Speed (f) / -	Adjust the set speed value
OUT_SP_12	Temperature safety limit (f) / -	Setting WD safety limit temperature with set value echo
OUT_SP_42	Speed safety limit (f) / -	Setting WD safety limit speed with set value echo
OUT_WD1	Watchdog time (f) / -	Setting the Watchdog mode 1
OUT_WD2	Watchdog time (f) / -	Setting the Watchdog mode 2
IN_PV_1	- / Sensor value (f)	Read actual external sensor value
IN_PV_2	- / Hotplate sensor value (f)	Read actual hotplate sensor value
IN_PV_4	- / Stirring speed (f)	Read stirring speed value
IN_PV_5	- / Viscosity trend (f)	Read viscosity trend value
IN_SP_1	- / Rated temperature (f)	Read rated temperature value
IN_SP_3	- / Rated safety temperature (f)	Read rated set safety temperature value
IN_SP_4	- / Rated speed (f)	Read rated speed value

Table 2: IKA C-Mag HS 7 Magnetic Stirrer and Heater

The IKA C-Mag HS 7 magnetic stirrer and heater exposes a simple programmable API to control its motor (which is used for stirring) and heater (table 2). Commands START_1 (START_4) and STOP_1 (STOP_4) are used to start and stop the heater (motor), respectively. Commands SET_MODE_m (where m = A, B, or D) and RESET allow switching between different operating modes. Commands IN_* and OUT_* are used to get and set different parameters, respectively. While both OUT_WD1 and OUT_WD2 set the watchdog time, they enable different watchdog modes.

Like the IKA module, the Tecan Cavro XLP 6000 syringe pump also exposes a simple API (table 3). Commands R and T are used to start and stop the pump; w and W are used to home the valves and the plunger; Z is used to set the homing position; g and G are used to start and stop the batch command loop; O and I are used to switch the valve to a set or given position; U, K, V, L, A, P, and D are used to set different parameters; and Q, ?, ?1, ?2, ?3, and ?6 are used to get different parameters.

Command	Arguments (type) / Responses (type)	Description
R	- / -	Start the pump
T	- / -	Stop the pump
w	- / -	Home the valves
W	Plunger home speed (i) / -	Home the plunger
Z	Plunger home speed (i) / -	Set the homing position
g	- / -	Start the batch command loop
G	Iteration count (i) / -	Stop the batch command loop
O	- / -	Set the valve to a set position
I	Position (i) / -	Set the valve to a given position
U	Pump Configuration (i) / -	Set pump configuration
K	Dead volume (i) / -	Set dead volume of pump
V	Velocity (i) / -	Set velocity of pump
L	Slope code (i) / -	Set slope code of pump
A	Position in counts (i) / -	Set position in counts
P	Distance in counts (i) / -	Set distance in counts
D	Distance (opp. direction) in counts (i) / -	Set distance in counts when moving in opposite direction
Q	- / Pump status (i)	Get pump status
?	- / Position in counts (i)	Get the current position in counts
?1	- / Start speed (i)	Get the current start speed
?2	- / Default speed in counts/s (i)	Get the default velocity in counts/s
?3	- / Cutoff speed (i)	Get the current cutoff speed
?6	- / Valve position (i)	Get the current valve position

Table 3: Tecan Cavro XLP 6000 Syringe Pump

The *ArduinoAugmentedQuantos* class controls both the Mettler Toledo Quantos system for powder dosing and the Aurduino-controlled stepper motor that is tasked with its z-axis control. Unlike the IKA and Tecan modules above, the Quantos module can be configured in many different ways, as evident from the commands shown in tables 4 and 6. The commands are self-explanatory (a positive consequence of tracing a higher-level class), and each command maps to a unique device-level command. Note that we omit a range of bookkeeping parameters such as user and sample IDs, and calibration and expiry dates; while these are returned as part of head_data and sample_data APIs, they are not currently being used by . 's data processing module therefore also filters these out.

Command	Arguments (type) / Responses (type)	Description
set_home_direction	Direction (i) / -	Sets the home direction
move_z_stage	Steps (i) / - Speed (f) / -	Moves the z stage in specified steps
home_z_stage	- / - *	Homes the z stage

Table 4: Arduino-Controller Stepper Motor

Next, we present the robot arm APIs. We start with the N9, which is a four-axis robot arm from North Robotics. The N9 is controlled via its C9 controller. table 5 lists a subset of the C9 APIs that is used to control the N9. Many commands are applied separately to each axis and therefore take a list of target axes as an argument. Commands BIAS, JLEN, and SPED are used to set different parameters; SPED also returns the value of parameters set by the robot. POS gets the position of the gripper or the end effector. ARM is the main command used to move the robot arm to the specified position; it also returns the final position after movement. Note that a position defines a point in the 3D space using a vector of three values.

Command	Arguments (type) / Responses (type)	Description
HOME	Axes (i $\times$ 4) / - Home only if needed (b) / - Skip actual homing (b) / -	Homes the N9 robot
HALT	Axes (i $\times$ 4) / -	Halts the given axes
BIAS	Elbow bias (i) / -	Sets the elbow bias
JLEN	Length (f) / -	Sets the elbow length
SPED	Default velocity (i) / Default velocity (i) Default acceleration (i) / Default acceleration (i)	Returns the velocity and acceleration of the N9 robot
POS	- / Position (f $\times$ 3)	Gets the position of the gripper or the end effector
MVNG	Axes (i $\times$ 4) / Moving States (b $\times$ 4)	Returns a list of moving statuses for the given axes
ARM	Position (f $\times$ 3) / Position (f $\times$ 3) Gripper position (f) / - Relative move (b) / - Velocity (f) / - Acceleration (f) / - Elbow bias (i) / -	Moves the robot arm to the specified position

Table 5: North Robotics N9 Robot

Command	**Arguments (type) / Responses (type)**	**Description**
start_dosing	- / -	Starts dosing of the solid
stop_dosing	- / -	Stops dosing of the solid
lock_dosing_pin_pos	- / -	Locks the dosing head to the dosing unit [sic]
unlock_dosing_pin_pos	- / -	Unlocks the dosing head from the dosing unit [sic]
set_pan_empty	- / -	Tells the Quantos that the pan is empty
tap_before_dosing	Yes/no (b) / -	Tap before dosing enabled
tap_while_dosing	Yes/no (b) / -	Tap during dosing enabled
tapper_intensity	Intensity (i) / -	Sets tapper intensity
tapper_duration	Seconds (i) / -	Sets tapper duration in seconds
target_mass	Milligrams (f) / -	Sets target mass in milligrams
tolerance_value	% (f) / -	Sets the tolerance value for powder dosing in percent
tolerance_mode	+/- or 0/+ (b) / -	Sets tolerance mode (+/- or 0/+)
dosing_algorithm	M/P/H (i) / -	Powder dosign algorithm (standard or advanced)
antistatic_enabled	Yes/no (b) / -	AntiStatic kit enabled
front_door_position	- / Position (i)	Returns the front door position
front_door_position	Position (i) / -	Sets the front door position
sampler_position	- / Position (i)	Get sampler position
sampler_position	Position (i) / -	Sets sampler position
sampler_status	- / On/off (b)	Returns sampler status
weigh_pan_status	- / Empty or not (b)	Returns the status of the weighing pan
head_data	- / Leveled (b) / Dose limit (i) / No. of doses (i) / Remaining quantity (f)	Returns the contents of the RFID chip of the dose head
sample_data	- / Quantity (f) / Target quantity (f) / Tolerance % (f) / Validity (b) / Accuracy % (f) / Seconds (f) / Tap while dosing (b) / Tap intensity (i) / Dosing mode (b) / Leveled (b) / Dose limit (i) / No. of doses (i) / Quantity left (f)	Returns the results data of the last dispense

Table 6: Mettler Toledo Quantos Powder Dosing System

Command	Arguments (type) / Responses (type)	Description
init	Velocity (f) / - Max. velocity (f) / - Joint velocity (f) / - Max. joint velocity (f) / - Gripper velocity (f) / - Gripper force (f) / -	Initializes the robot arm
default_joint_velocity	Joint velocity (f) / Joint velocity (f)	Sets and gets the default joint velocity
pose	- / TCP position (f $\times$ 6)	Gets the pose of the robot arm
joint_positions	- / Joint positions (f $\times$ 6)	Gets the joint positions of the robot arm
joint_count	- / Joint count (i)	Gets the joint count
tool_offset	Tool offset location (f $\times$ 6)/ -	Sets the tool offset
tool_mass	Payload (f) / -	Sets the tool mass
gripper_position	- / Gripper position (f)	Gets the gripper position
stop	- / -	Stops the robot arm
emergency_stop	- / -	Emergency stop of the robot arm
wait	Timeout (f) / - Wait for running (b) / -	Waits for the robot arm to start or stop and timeouts after a while
move_to_location	Location (f $\times$ 6) / - Base reference frame (f $\times$ 6)/ - Velocity (f) / - Acceleration (f) / - Relative (b) / - Wait (b) / - Timeout (f) / -	Move robot arm to the specified location
move_joints	Joint positions (f $\times$ 6) / - Velocity (f) / - Acceleration (f) / - Relative (b) / - Wait (b) / - Timeout (f) / -	Moves the joints to the specified location
move_circular	Midpoint location (f $\times$ 6) / - End location (f $\times$ 6) / - Base reference frame (f $\times$ 6)/ - Velocity (f) / - Accelration (f) / - Wait (b) / - Timeout (f) / -	Moves the arm in a circular direction
open_gripper	Position (f) / - Force (f) / - Velocity (f) / - Wait (b) / - Timeout (f) / -	Opens the gripper
wait_for_gripper_stop	Timeout (f) / -	Waits for the gripper to stop and timeouts after a while

Table 7: Universal Robots UR3e Robot Arm

Unlike the N9, the is connected over TCP, via a third-party Python package *urx*. As a result, traces the methods exposed by 's *class UR3Arm(RobotArm)* rather than the device-level commands that are exposed by the robot itself. Nonetheless, the dataset (table 7) resembles the N9 dataset in terms of the basic actuation APIs and is therefore equally useful. For convenience, the APIs use both cartesian and polar coordinate systems. Each *frame* or *location* in the space is therefore represented using two tuples: (x, y, z) and (rx, ry, rz), i.e., , six floats in total. This representation is used for denoting both joint positions and tool center point (TCP). 's data processing filters out all connection related parameters that happen to be a part of the argument list for many methods in *class UR3Arm(RobotArm)*.

2 Power Dataset

Name	**Description**
target_q	Target joint positions
target_qd	Target joint velocities
target_qdd	Target joint accelerations
target_current	Target joint currents
target_moment	Target joint moments
target_TCP_pose	Target coordinates
target_TCP_speed	Target speed
actual_execution_time	thread execution time
actual_tool_accelerometer	X, Y, Z accelerometer values
speed_scaling	Speed scaling of trajectory limiter

Table 8: Power Monitoring Features

3 Experiment Procedure Steps

#	Description
1.	The N9 robot picks up a vial from the deck and places it on the on-deck gripper.
2.	The N9 robot picks up the vial.
3.	The Quantos door opens.
4.	The N9 robot places the vial inside the quantos.
5.	The Quantos measures the weight.
6.	The N9 robot removes vial from the Quantos and places it on the on-deck gripper.
7.	The Quantos door closes.
8.	The N9 robot uncaps the vials and places the cap on the on-deck caper.
9.	The N9 robot picks up the vial.
10.	The Quantos door opens.
11.	The N9 robot places the vial inside the quantos.
12.	The Arduino Stepper Motor moves down towards the vial.
13.	The Quantos door closes.
14.	The Quantos starts the dosing of the solid.
15.	Once dosing is done, the Arduino Stepper Motor moves the dosing head up.
16.	The Quantos door opens.
17.	The N9 robot picks up the vial.
18.	The Quantos door closes.
19.	The N9 robot places the vial on the on-deck gripper.
20.	The N9 robot moves the slider where the vial is going to be placed.
21.	The N9 robot picks up the vial and places on the magnetic stirrer.
22.	The N9 robot closes the slider.
23.	The N9 robot goes to the needle station, picks up the needle and removes the lid.
24.	The N9 robot moves to the liquid dosing station.
25.	The Tecan Cavro pumps the liquid.
26.	The N9 robot moves to the vial.
27.	The Tecan Cavro dispenses liquid inside the vial.
28.	The Magnetic Stirrer starts.
29.	The Magnetic Stirrer stirs the mixture of the vial.
30.	The camera on the deck takes a picture of the vial to check the turbidity of the solution.
31.	The steps from 24 to 30 are repeated until the mixture is dissolved.
32.	Once this is done, the cleanup vial is called.
33.	The N9 robot arm discards the needle.
34.	The N9 robot arm picks the vial.
35.	The Quantos door opens.
36.	The N9 robot places the vial inside.
37.	The Quantos measures the weight.
38.	The N9 robot picks the vial from Quantos.
39.	The N9 robot places the vial solution on the vial tray

Table 9: Procedure: Automated Solubility

#	Description
1.	The N9 Arm picks up a tube with a filter from the deck and opens its cap.
2.	The N9 Arm removes the filter from the tube and puts the filter in the filter holder.
3.	The N9 robot picks up the tube.
4.	The Quantos door opens.
5.	The N9 robot places it inside the quantos.
6.	The Arduino Stepper Motor moves the dosing head down.
7.	The Quantos door closes.
8.	The Quantos starts dosing of the solid.
9.	Once dosing is done, Arduino Stepper Motor moves the dosing head up.
10.	The Quantos door opens.
11.	The N9 Arm picks up the tube and moves out of Quantos.
12.	The Quantos door closes.
13.	The N9 Arm places the tube in vial tray, removes filter from holder and places the filter in tube
14.	The N9 Arm goes to the needle station, picks up the needle and removes the lid.
15.	The N9 Arm moves to the liquid dosing station.
16.	The Tecan Cavro pumps the liquid.
17.	The N9 Arm moves to the tube.
18.	The Tecan Cavro dispenses liquid inside the tube.
19.	The N9 Arm closes the tube cap and transfers it to the Magnetic Stirrer.
20.	The Magnetic Stirrer stirs the solution.
21.	The N9 Arm picks up the tube and puts it on the mini tube tray.
22.	The N9 Arm moves away.
23.	The UR3 Arm picks up the cap of the centrifuge and places the cap on the deck.
24.	The camera checks the centrifuge position.
25.	The UR3 Arm picks up a rod, adjusts centrifuge position and places the rod back.
26.	The UR3 Arm then picks up the vial and puts it inside the centrifuge.
27.	The Centrifuge then puts the cap back.
28.	The Centrifuge is started to filter the solid from liquid.
29.	Once this is done, the UR3 Arm picks up the cap and places it on the deck.
30.	The Camera checks the centrifuge position, picks up the rod, adjusts centrifuge position and places the rod back.
31.	The UR3 Arm picks up the tube from the Centrifuge and places it on the station.
32.	The UR3 Arm places the cap back.
33.	The N9 Arm picks up the tube, opens the tube cap and sends the tube to the vial tray.
34.	The N9 Arm picks up the new tube from the tube tray and uncaps the new tube.
35.	The N9 removes the filter from the old tube, sends the filter to the new tube.

Table 10: Procedure: Crystal Solubility Profiling

#	Description
1.	The N9 robot picks up a vial from the deck and places it on the on-deck gripper.
2.	The N9 robot picks up the vial.
3.	The Quantos door opens.
4.	The N9 robot places the vial inside the quantos.
5.	The Quantos measures the weight.
6.	The Quantos Balance checks the solid inside the quantos
7.	If it is different from the one that is required for the experiment, the solid is replaced.
8.	To replace the solid, the UR3 robot moves to the Quantos.
9.	UR3 robot removes the solid tube and places it to the on-wall RFID positions of the tubes.
10.	UR3 robot picks up the correct one and moves towards the Quantos.
11.	UR3 robot fixes the correct solid tube inside the Quantos.
12.	The vial is removed from the Quantos and placed on the on-deck gripper.
13.	The Quantos door closes.
14.	The N9 robot uncaps the vials and places the cap on the on-deck caper.
15.	The N9 robot picks up the vial.
16.	The Quantos door opens.
17.	The N9 robot places the vial inside the quantos.
18.	The Arduino Stepper Motor moves down the dosing head.
19.	The Quantos door closes.
20.	The Quantos starts the dosing of the solid.
21.	Once dosing is done, the Arduino Stepper Motor moves the dosing head up.
22.	The Quantos door opens.
23.	The N9 robot picks up the vial.
24.	The Quantos door closes.
25.	The N9 robot places the vial on the on-deck gripper.
26.	The N9 robot moves the slider where the vial is going to be placed.
27.	The N9 robot picks up the vial and places on the magnetic stirrer.
28.	The N9 robot closes the slider.
29.	The N9 robot goes to the needle station, picks up the needle and removes the lid.
30.	The N9 robot moves to the liquid dosing station.
31.	The Tecan Cavro pumps the liquid.
32.	The N9 robot moves to the vial.
33.	The Tecan Cavro dispenses liquid inside the vial.
34.	The Magnetic Stirrer starts.
35.	The Magnetic Stirrer stirs the mixture of the vial.
36.	The camera on the deck takes a picture of the vial to check the turbidity of the solution.
37.	The steps from 31 to 36 are repeated until the mixture is dissolved.
38.	Once this is done, the cleanup vial is called.
39.	The N9 robot arm discards the needle.
40.	The N9 robot arm picks the vial.
41.	The Quantos door opens.
42.	The N9 robot places the vial inside.
43.	The Quantos measures the weight.
44.	The N9 robot picks the vial from Quantos.
45.	The N9 robot places the vial solution on the vial tray.

Table 11: Procedure: Automated Solubility with UR3 Arm