Very basic SPEC commands

Donald A. Walko
Beamline 7ID, Advanced Photon Source, Argonne National Laboratory, Argonne, IL 60439
d-walko@anl.gov

January 18, 2016

This document presents the typical ways to use some basic SPEC commands for a single-crystal diffraction experiment. An online manual can be found at `http://www.certif.com` and many topics are described in the help utility. SPEC is a (case-sensitive) command-line based program. Therefore it is important to know the correct commands; fortunately, some commands will list the type of parameters needed if the wrong type (or number) are entered.

"Information" and miscellaneous commands

`pa`
Lists parameters such as the orientation matrix, lattice parameters, operating mode, wavelength, etc.

`wh`
where; lists positions of the diffractometer motors, reciprocal lattice coordinates, and some relevant angles.

`wa`
where all; lists positions of all the SPEC motors (in user units and dial units).

`p` *some expression*
print; can be used to print a variable's value:
```
p F_CHI
```
or as a calculator:
```
p 5*sin(PI/4)
```

`help` [*topic*]
calls SPEC's help utility

`ct` [*time*]
counts and lists results for all scalers. The counting is for *time* sec or for the default time if *value* is omitted (such as 1 sec). But if *value* is negative, then counting continues until the monitor (a particular scaler) reaches *value* counts. Examples of use:
```
ct
ct 10
ct -100000
```

~~`startup`~~
~~Initializes a variety of parameters by calling the macros `newsample`, `newfile`, `setscans`, `setplot`, and `setgeo`. These semi-self-explanatory macros can also be called directly.~~

`quit`
quit out of SPEC.

Simple motor motions

`umv` *motor_name position*
move *motor_name* to absolute *position* (in user units). Examples:
```
umv th 20
umv th CEN
```
See below under `dscan` for information on the variable `CEN`.

`umvr` *motor_name rel_position*
move *motor_name* by the relative amount *rel_position* from its current position.

`ubr` *H K L*
move motors to the reciprocal lattice point (Bragg point) defined by the Miller indices *H K L*

`ca` *H K L*
calculate the motor position for the reciprocal lattice point *H K L*. It's a good habit to do `ca` before `ubr`, to avoid unexpected motor motions.

`tw` *motor_name delta*
tweak; interactive subroutine to move *motor_name* by *delta*. Once in the subroutine, each time you hit `Enter` the motor moves by *delta*. You can change direction with p/n or +/-, and also change *delta* by entering a new value. Escape by hitting `CTRL-C`.

Basic scans

For scans in SPEC, one enters the number of intervals, which is one more than the number of points. Thus, the step size is (ending point) - (starting point)/intervals. The unit of time is seconds per point if positive, or monitor counts per point if negative.

`loopscan` *npts [count_time [sleep_time]]*
time-lapse scan: sit and count for *npts* points without moving motors

`timescan` *[count_time [sleep_time]]*
indefinite time-lapse scan

Motor scans

`ascan` *motor_name start end intervals time*
absolute scan: *motor_name* starts at *start* and ends at *end* (in user units). At the end of the scan, *motor_name* stays at *end*. Example:
```
ascan th 5 7 30 1
```

`dscan` *motor_name rel_start rel_end intervals time*
relative (differential) scan: *motor_name* starts at *start + current_position* and ends at *end + current_position*. At the end of the scan, *motor_name* returns to its previous position. This is the same as a `lup` (line up) scan. Example:
```
dscan th -1 1 30 1; umv th CEN
```
The variable `CEN` is calculated after each scan, and is the absolute position of the peak's center. Note that if you typed **`umvr th CEN`** or **`umv phi CEN`** you could get into big trouble! It may also give weird results if there is no peak, or if the FWHM couldn't be calculated from the scan.

`a2scan` *motor_name1 start1 end1 motor_name2 start2 end2 intervals time*
absolute scan of two motors: *motor_name1* starts at *start1* and ends at *end1*, while *motor_name2* starts at *start2* and ends at *end2*. `a3scan` and `a4scan` operate similarly, `d2scan`, `d3scan`, and `d4scan` are multimotor relative scans.

`mesh` *motor_name1 start1 end1 intervals1 motor_name2 start2 end2 intervals2 time*
motor mesh scan. A scan of *motor_name1* is done for each point of *motor_name2*, all of which is stored as one SPEC scan. Example:
```
mesh th 5 7 50 tth 10 14 30 1
```
In this example, the full scan contains 51*31=1581 points.

Reciprocal space scans

`hscan` *h_start h_end intervals time*
linear scan in reciprocal space along the H axis. The values of K and L during this scan are based on the previous position in reciprocal space, so you may need to use the `ubr` command to first move to the appropriate point. Example:
```
ubr 1 1 1; hscan .9 1.1 20 2
```

`kscan` *k_start k_end intervals time*
same as `hscan` but along the K axis.

`lscan` *l_start l_end intervals time*
same as `hscan` but along the L axis.

`hklscan` *h_start h_end k_start k_end l_start l_end intervals time*
linear scan in reciprocal space along a general direction. For example, if you wanted to scan in some direction along H and K thru the (111) Bragg peak:
```
hklscan .9 1.1 1.2 0.8 1 1 20 1
```

`hklmesh` *Q1 start1 end1 intervals1 Q2 start2 end2 intervals2 time*
reciprocal space mesh scan. *Q1* and *Q2* are literally `H`, `K`, or `L`. Thus this type of mesh scan is limited to be along the principal axes of reciprocal space. The value of the third reciprocal space coordinate during this scan is based on the previous position in reciprocal space, so you may need to move there first. For example, if you wanted to scan in the *H-L* plane thru the (111) Bragg peak:
```
ubr .8 1 .9; hklmesh H .8 1.2 20 L .9 1.1 20 1
```

Orientation matrix commands

With two nonparallel reflections entered into the orientation matrix, SPEC can perform the transformation between diffractometer angles and reciprocal lattice coordinates.

`or0`
Sets the primary reflection, when you are at the appropriate angles.

`or1`
Sets the secondary reflection, when you are at the appropriate angles.

`setor0`
Sets the primary reflection by asking for the reflection's angles.

`setor1`
Sets the secondary reflection by asking for the reflection's angles.

`setlat`
Command to input the lattice parameters.

A few useful variables

`LAMBDA`: wavelength in Ångstroms

~~`OMEGA`: *theta-(twotheta)/2*~~

`ALPHA`: incident angle (typically)

`BETA`: exit angle (typically)

`AZIMUTH`: rotation angle of reference vector about scattering vector (may vary with geometry)