SUPPLEMENTARY INFORMATION for:

Development of a Raman spectroscopy system for in situ monitoring of microwave-assisted inorganic transformations

John Jamboretz[1], Andreas Reitz[1], and Christina S. Birkel[1,2*]

[1] School of Molecular Sciences, Arizona State University, Tempe AZ-85282, USA.
[2]Technische Universität Darmstadt, 64287 Darmstadt, Germany

Furnace thermocouple to sample thermocouple offset

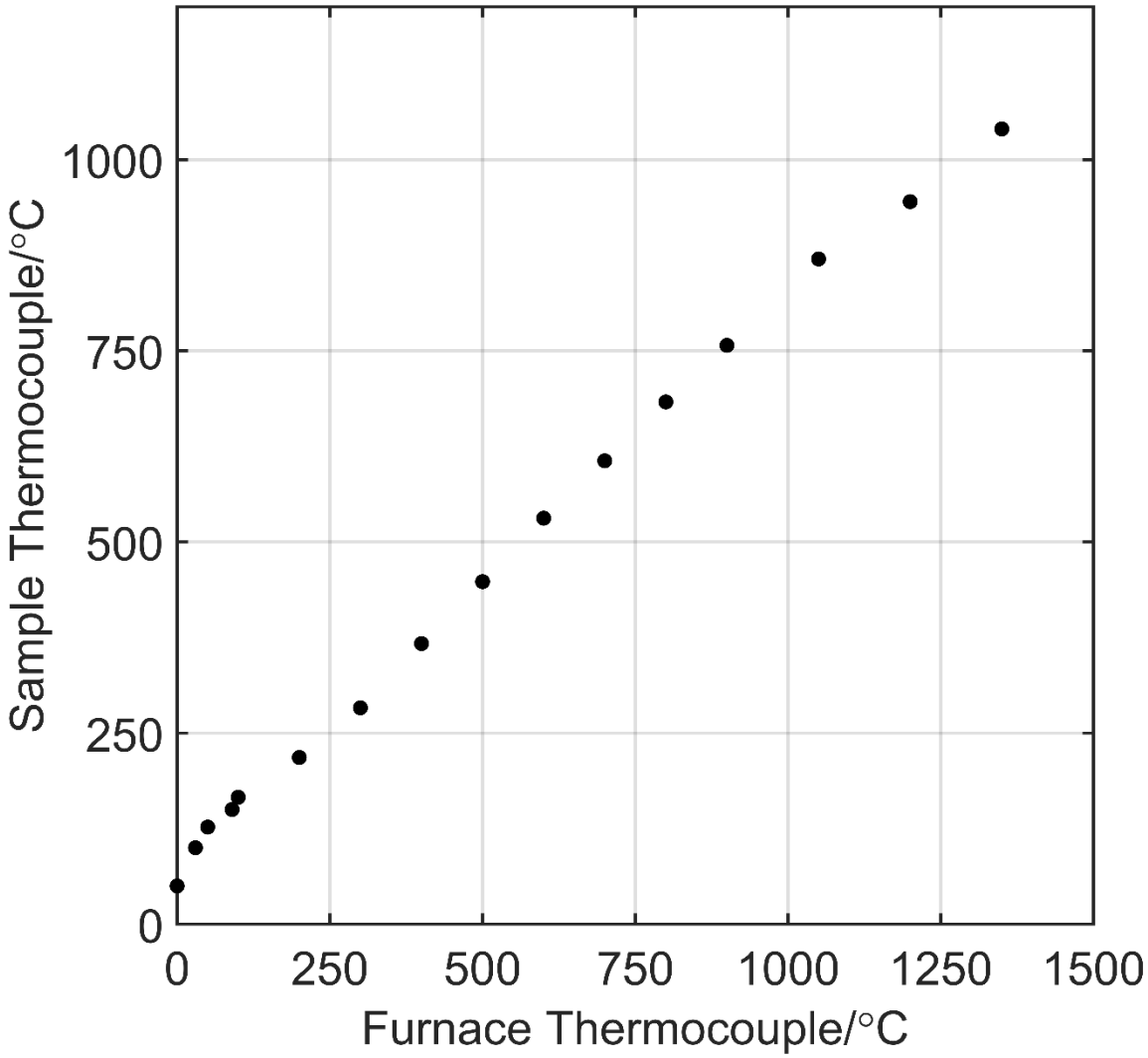

Fig. S1: Measured temperature at the sample with the “sample thermocouple” at given equilibrated set furnace temperatures as measured by the “furnace thermocouple”.

Example Raman spectra

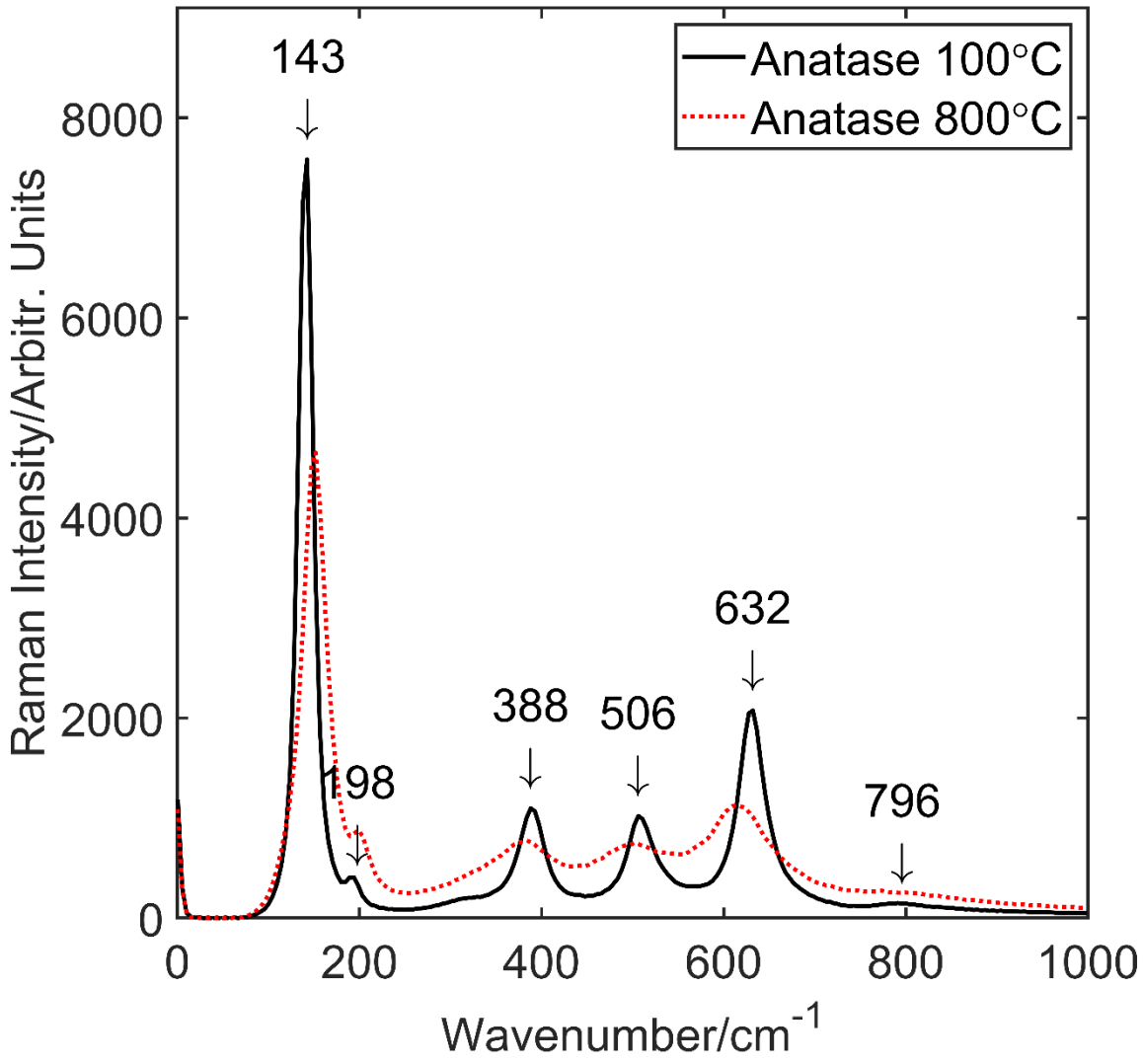

Fig. S2: Example Raman spectra of anatase TiO_2 w/ 1 mol% CuO at 100 °C (black solid line) and 800 °C (red dotted line). Annotations indicate the peak positions at 100 °C for the six identified Raman modes.

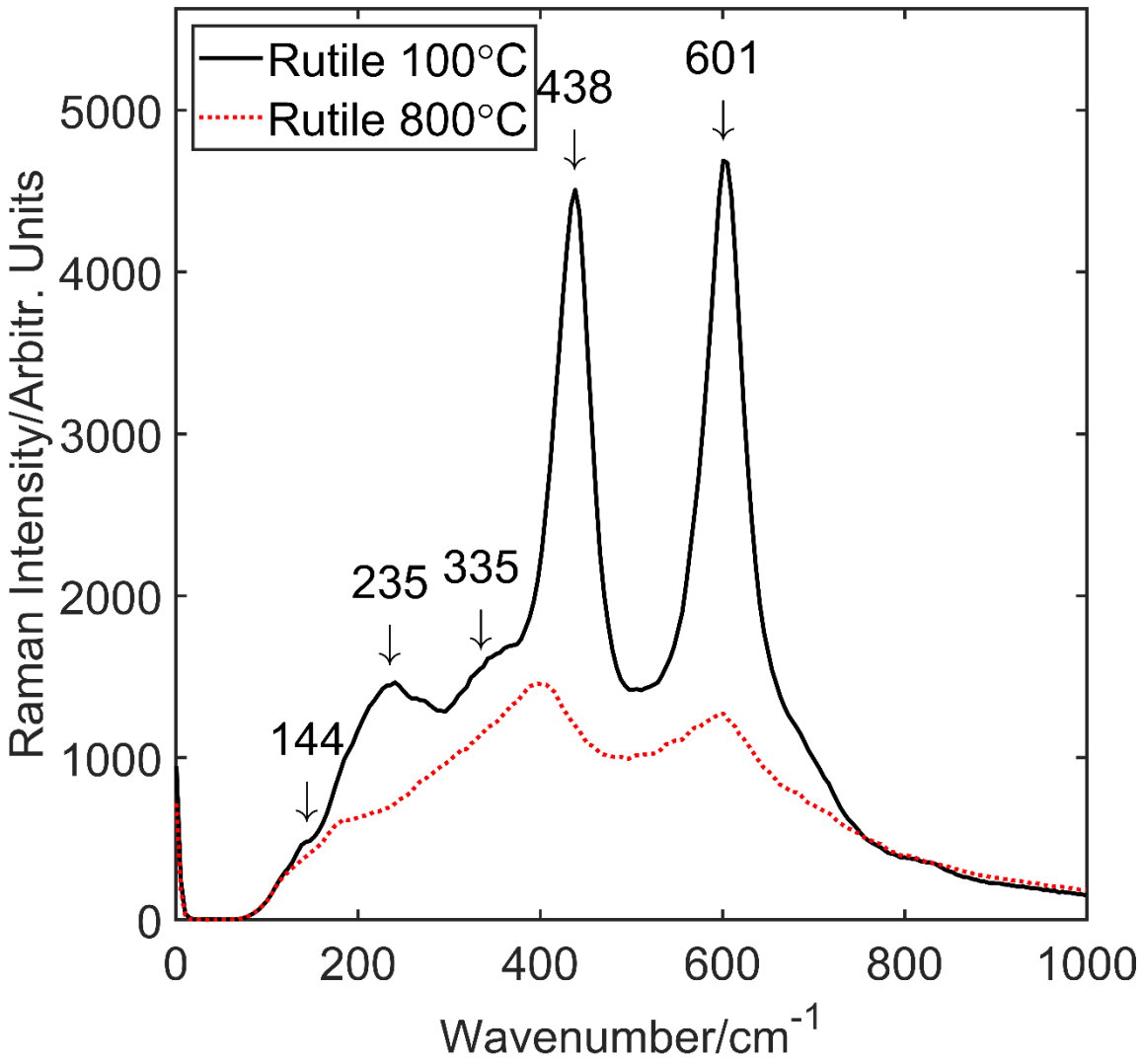

Fig. S3: Example Raman spectra of rutile TiO_2 w/ 1 mol% CuO at 100 °C (black solid line) and 800 °C (red dotted line). Annotations indicate the peak positions at 100 °C for the four identified Raman modes.

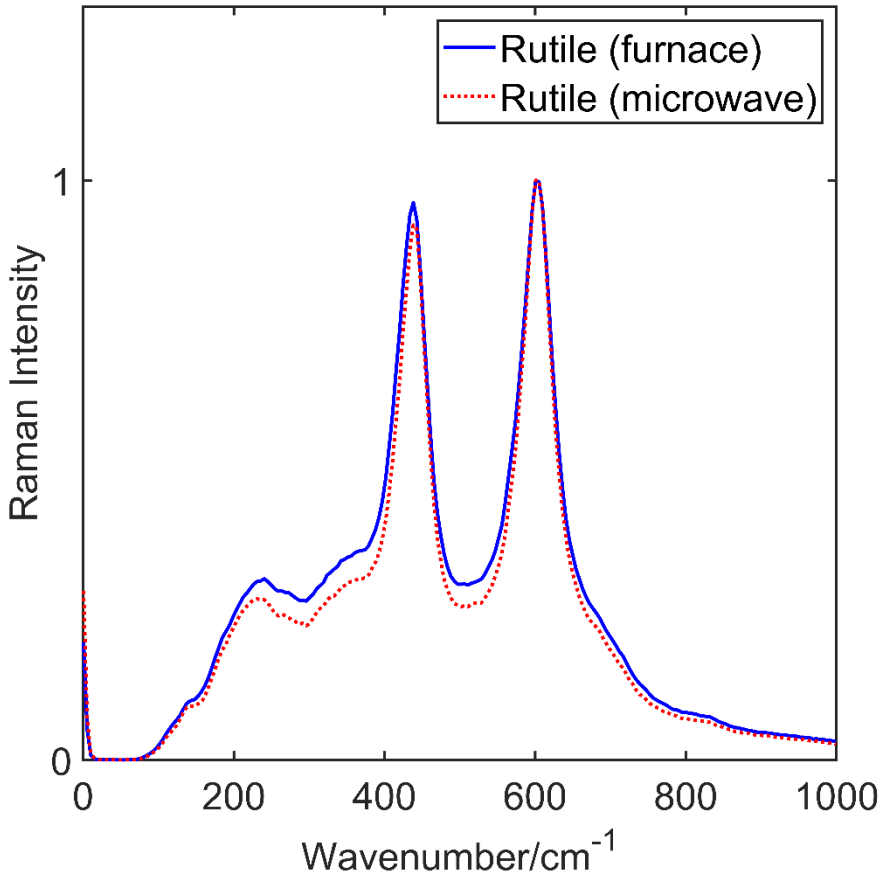

Fig. S4: Raman spectra of the rutile samples derived from the furnace (blue solid line) and microwave (red dotted line) heating methods. The intensities of each are normalized.

Powder X-Ray Diffraction (PXRD):

X-Ray powder diffractograms were collected on a Bruker D2 Phaser (2nd Generation) with a Cu Kα (λ = 1.54184, 30kV, 10mA) source and a Ni filter. Starting material (anatase TiO_2 with 1 mol% CuO) and products were thoroughly ground in an agate mortar before loading on a single-crystal Si stage. Collections were performed at room temperature in a 2θ range of 7-90 ° and a step size of .099 °/step. The total measurement time is approximately 13 minutes.

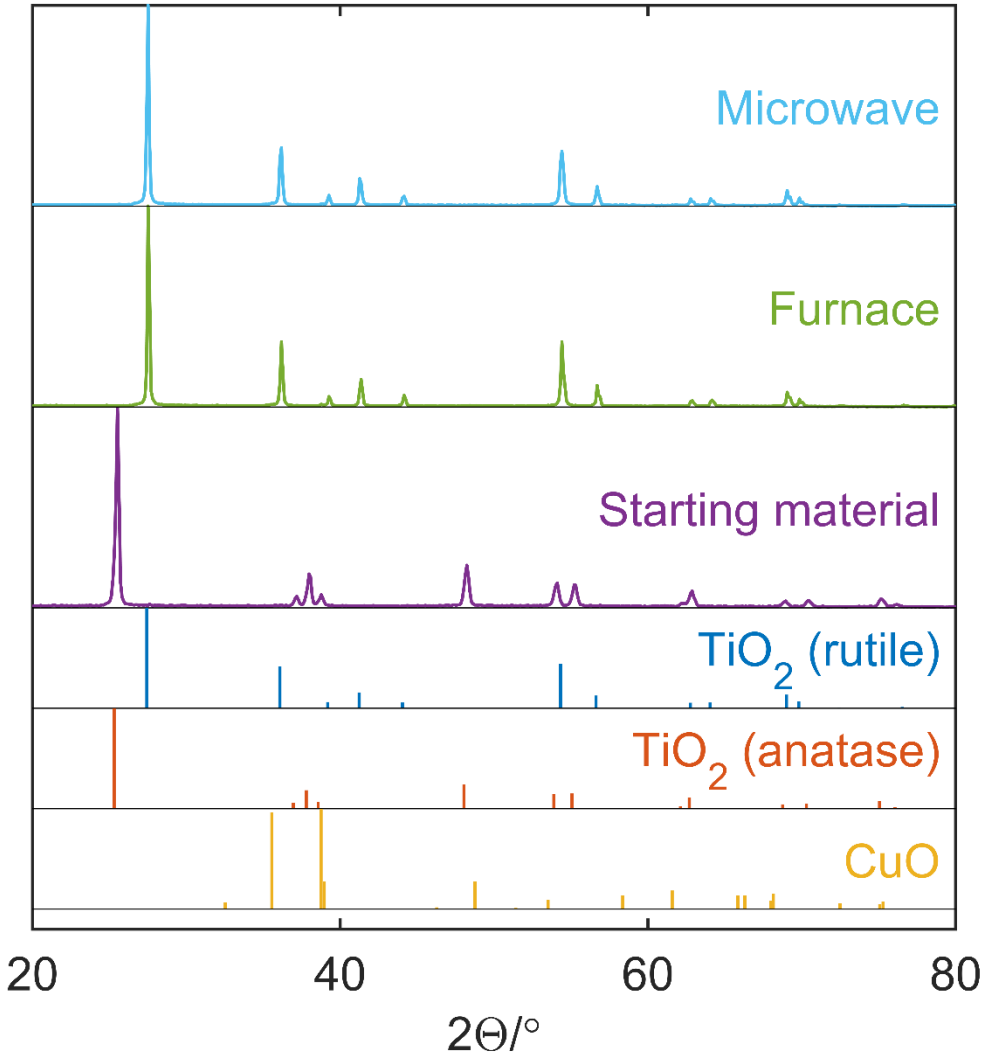

Fig. S5: PXRD data for the starting material, furnace product, and microwave product. Indexed reflections for TiO_2 (rutile)[1], TiO_2 (anatase)[2], and CuO[3].

Transition change point analysis:

The transition change point analysis was performed by calculating the integrated peak area by the trapezoidal method with linear baseline subtraction for selected wavenumber ranges for each peak analyzed. The selected ranges are summarized in Table S1:

Peak	**P1 (cm^{-1})**	**P2 (cm^{-1})**
A-143	62	189
A-506	452	543
A-632	552	734
R-393	374	493
R-592	623	752

Table S1: Selected wavenumber ranges for peaks analyzed. Peaks are denoted by their phase (A: anatase; R: rutile) and measured positions in wavenumbers (cm^{-1}) at 100 °C. P1 and P2 are the bounds for each peak in wavenumbers (cm^{-1}).

The evolution of the peak area was tracked for selected time ranges summarized in Table S2:

Phase (experiment)	**T1 (s)**	**T2 (s)**
Anatase (furnace)	2800	3370
Rutile (furnace)	3180	3900
Anatase (microwave)	1800	2240
Rutile (microwave)	2100	2420

Table S2: Selected time ranges for analyzed peaks in each phase for each experiment denoted by their phase and experiment. T1 and T2 are the time bounds for each in seconds.

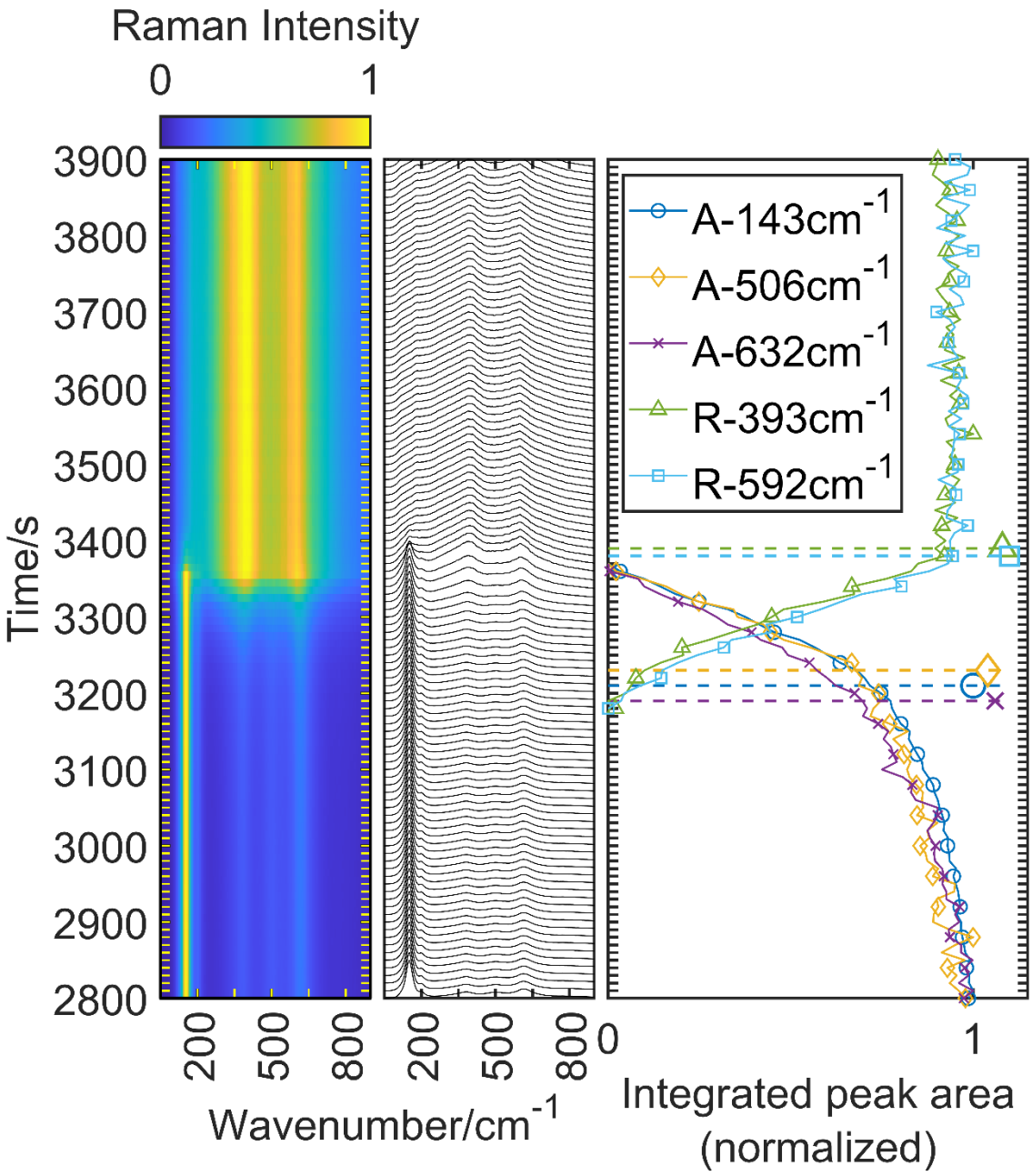

Fig. S6: Transition change point analysis for the furnace experiment. Left plot: Intensity surface plot (normalized intensities for each time step) for the time range of the transition change point analysis. Middle plot: Individual spectra at each time step (normalized intensities for each time step) on the same time scale. Right plot: Normalized evolution of the integrated peak area (solid lines) and calculated linear changepoints for each (dashed horizontal lines) on the same time scale. Selected peaks are denoted by their phase (A: anatase; R: rutile) and measured positions in wavenumbers (cm^{-1}) at 100 °C.

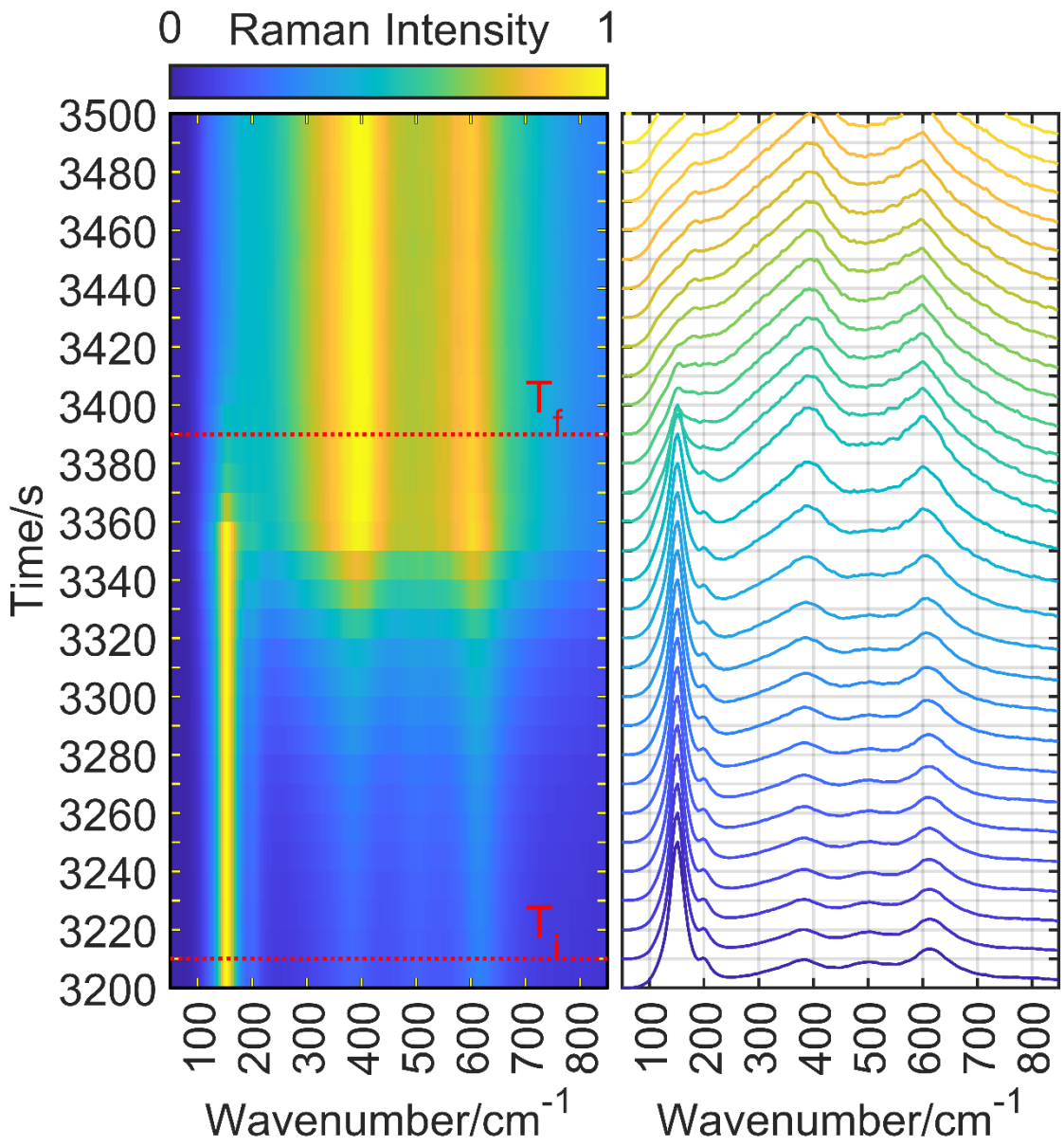

Fig. S7: Detail of Raman spectra evolution around the phase transition for the furnace experiment. Left plot: Intensity surface plot (normalized intensities for each time step) for a reduced time range including the transition. Dashed red lines indicate the calculated transition initiation time (labelled T_i) and transition completion time (labelled T_f) using the linear changepoint analysis. Right plot: Individual spectra at each time step (normalized intensities for each time step) on the same time scale (note: colors do not indicate intensities in the right plot).

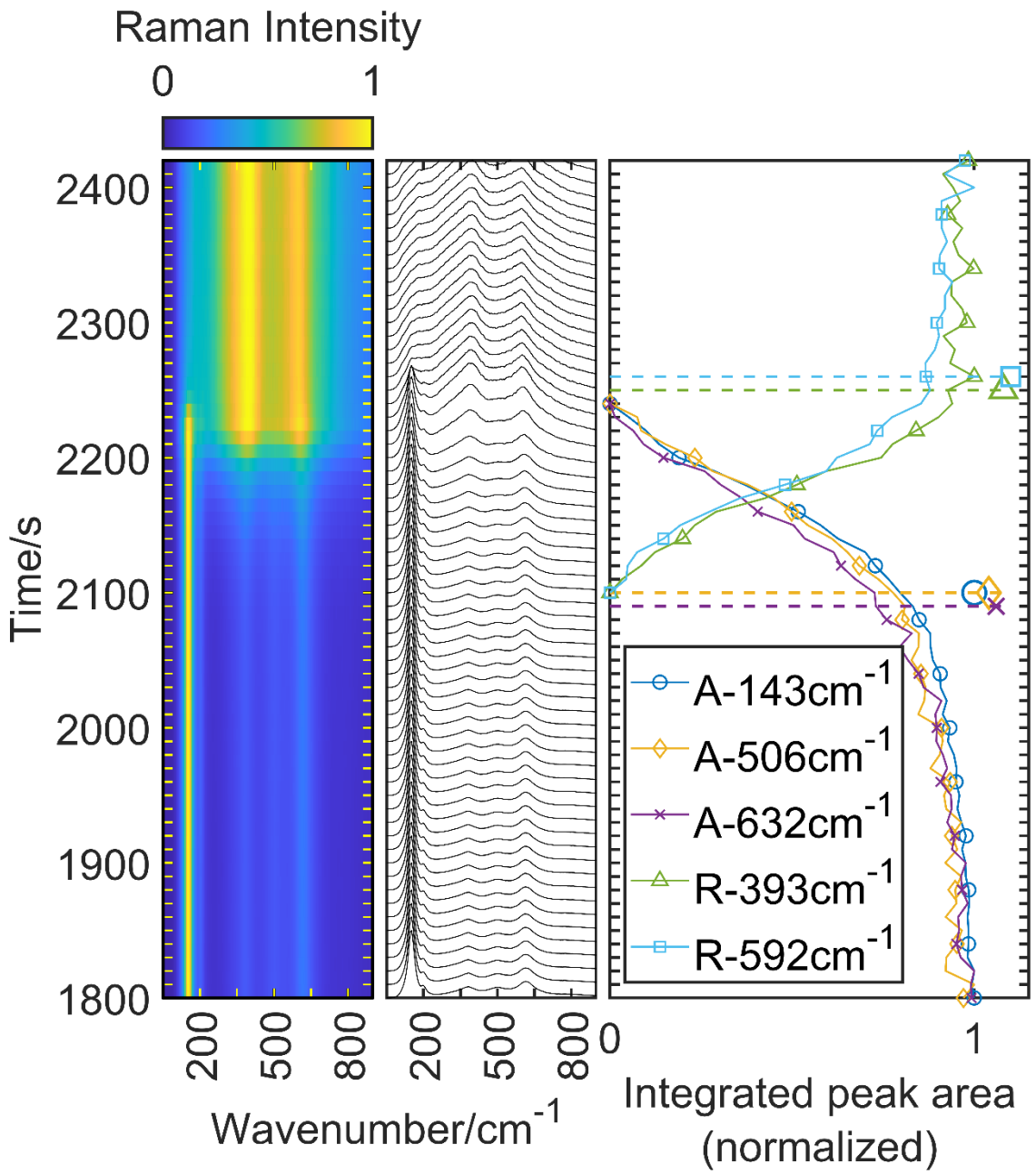

Fig. S8: Transition change point analysis for the furnace experiment. Left plot: Intensity surface plot (normalized intensities for each time step) for the time range of the transition change point analysis. Middle plot: Individual spectra at each time step (normalized intensities for each time step) on the same time scale. Right plot: Normalized evolution of the integrated peak area (solid lines) and calculated linear changepoints for each (dashed horizontal lines). Selected peaks are denoted by their phase (A: anatase; R: rutile) and measured positions in wavenumbers (cm^{-1}) at 100 °C.

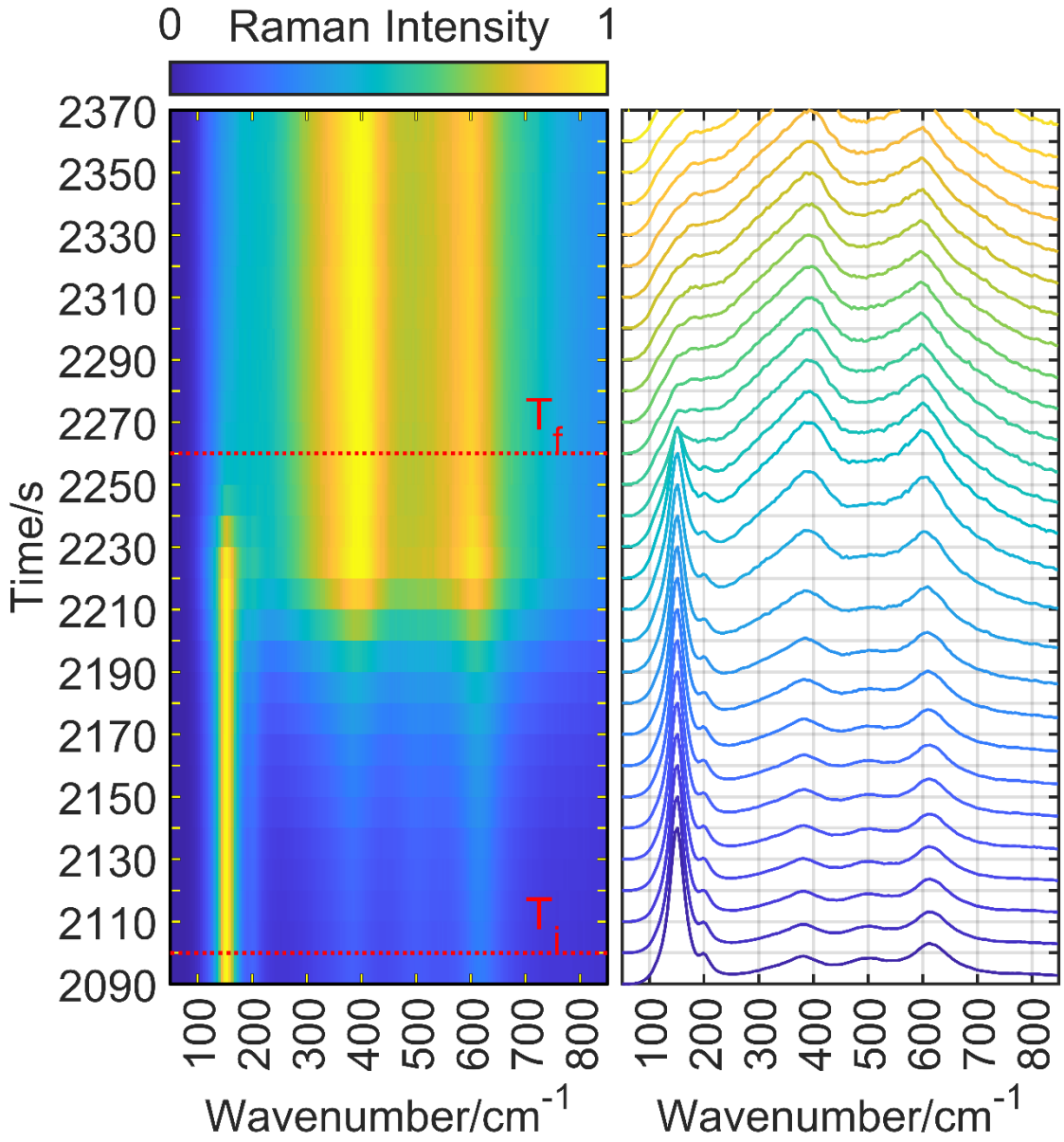

Fig. S9: Detail of Raman spectra evolution around the phase transition for the microwave experiment. Left plot: Intensity surface plot (normalized intensities for each time step) for a reduced time range including the transition. Dashed red lines indicate the calculated transition initiation time (labelled T_i) and transition completion time (labelled T_f) using the linear changepoint analysis. Right plot: Individual spectra at each time step (normalized intensities for each time step) on the same time scale (note: colors do not indicate intensities in the right plot).

Anatase 143 cm^{-1} peak position with time

The peak position was derived from fitting with an asymmetric pseudo-Voigt profile described elsewhere.[4]

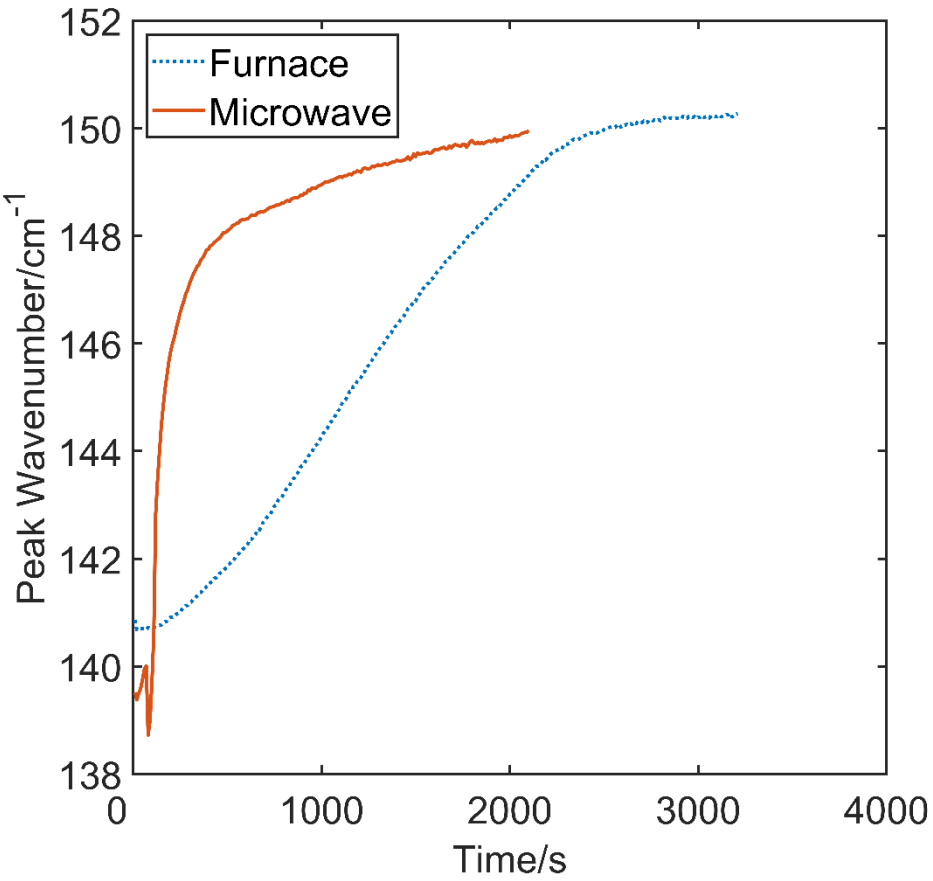

Fig. S10: Evolution of the peak position of the main anatase band (A-143 cm^{-1}) for the furnace (blue dotted line) and microwave (orange solid line) experiments. Note: The small dip at the beginning of the microwave experiment is attributed to evaporation of residual water from the sample/container obscuring the optical path. A large dip in Raman intensity corresponds exactly with this observed wavenumber shift. The microwave experiment is started at room temperature whereas the furnace experiment is started at 100 °C and is presumed to be dry which may explain why this is not observed during the furnace experiment.

References

[1] W. H. Baur, *Acta Crystallogr.* **1956**; *9*, 515–520.

[2] C. J. Howard, T. M. Sabine, F. Dickson, *Acta Crystallogr. Sect. B Struct. Sci.* **1991**; *47*, 462–468.

[3] S. Åsbrink, L. J. Norrby, *Acta Crystallogr. Sect. B Struct. Crystallogr. Cryst. Chem.* **1970**; *26*, 8–15.

[4] V. I. Korepanov, D. M. Sedlovets, *Analyst.* **2018**; *143*, 2674–2679.

MATLAB instrument control and data processing:

Instrument control:

Instrument control is implemented within a MATLAB program interfacing with the following instrument hardware components: the spectrometer, the thermocouple reader, and the raster control hardware.

Interfacing with the spectrometer was accomplished using the LightField software provided by the spectrometer supplier (Princeton Instruments) with Add-ons and Automation SDK installed. This allows MATLAB to initialize the Lightfield software, communicate the number of spectra to acquire, and return the data back to the MATLAB environment. The collection time of each spectra and the must be saved previously in a LightField experiment file. To achieve proper synchronization, the camera is triggered repeatedly at a specified frequency; in this work, 1Hz to achieve 1 spectrum/second. The exposure time must be adjusted to allow readout by the camera before the next trigger is received; for 1 spectrum/second it is set at 490ms.

Interfacing with the thermocouple acquisition device is achieved by using the MATLAB addon Data Acquisition Toolbox. This allows a serial connection to the USB thermocouple reader (NI) for triggering the reading of the thermocouple output and communication of the data back to the MATLAB environment.

Interfacing with the rastering control microcontroller is achieved by using native MATLAB serial communication abilities. This establishes a serial connection to the stepper microcontroller for sequential movement commands in the device. The microcontroller is configured with GRBL (https://github.com/grbl/grbl) which allows serial communication of G-code commands to be received, interpreted, and translated into commands to the stepper motor controllers. Patterns of any custom geometry can be achieved. The microcontroller must first be calibrated to achieve proper speed and dimensions based on the commands sent dependent on the hardware setup. Calibration was performed using Universal G-code Sender (https://winder.github.io/ugs_website/).

To begin an experiment the user enters the desired number of spectra to acquire (*numberacquires)* and the total length of each acquisition (*acquiretime)* into *startexperiment*. *startexperimentfunc* initializes Lightfield and sets the number of spectra to acquire.

The thermocouple collection hardware (NI) is then initialized. It is necessary to utilize the function *parfeval* within the Parallel Computing Toolbox to allow the data collection to run in the background. The function *thermocouplefunc* takes the number and time of acquisitions to determine how long and how often to record the temperature.

thermocouplefunc opens a serial connection with the thermocouple recorder and requests the temperature sequentially based on the number of acquires and acquire time requested. Rarely, the data communication can fail to complete within the time requested so a *try, catch* statement is included. In the event of a failed communication that entry is recorded as NaN (not a number) and the next collection is attempted. The initialization step can sometimes take longer than expected, so the actual collection start time is recorded just before reading data (*time1)* which can be used to correct for this, if desired. The typical error is on the order of ~10s and was deemed

insignificant in this study, so no correction was applied. The temperature data is returned as an array *thermocouplerecord*.

The rastering hardware is initialized within the *startexperimentfunc* by another parallel evaluation of *rasterfunc*.

This function takes the number of acquires and acquire time requested and calculates the total time to run the rastering pattern. First a serial communication is opened to the rastering microcontroller using *serialport* with the particular USB port utilized (e.g. COM1) and baud rate. Once this is complete, the initial time is recorded by *time0.* Then, the parameters for rastering speed (*scanspeed*) and orbital pattern dimensions (*innerradius, outerradius)* are entered which can be customized. The *pausetime* must be optimized such that each step in the pattern completes before the next step is communicated to avoid overloading the serial buffer but without delays to the next step. The parameters included here operate without delays between steps and without overload for over 2 days of collection time. An initial command moves the probe from the central storage position into the raster pattern. Then sequential commands are sent to move the probe in arcs to complete the orbital pattern. The elapsed time is recorded after each orbital cycle until the total experiment time is exceeded, at which time one final command returns the probe to the central storage position.

After temperature collection and rastering are started, the spectra acquisition is initiated in the LightField environment. Once complete the spectral data will be returned as a cell *rawspectraldata*. The LightField program is then closed. s*pectralaquirestarttime* logs the approximate start time for spectra acquisition.

The temperature data is returned as *thermocouplerecord* and the actual recording start time as *thermocouplestarttime*.

The timing error between temperature and spectra collection is recorded by *temperaturetospectraltimeoffset* where negative offset values indicate temperature collection was delayed relative to spectral collection. Finally, the *rawspectraldata* cell is reformatted into 2 arrays, *spectraldatareformat* and *backgroundatareformat*, in which rows are the pixels and columns are the time. The two background regions are summed.

spectraldata, backgrounddata, and *temperaturedata* outputted from *startexperimentfunc* can then be saved to the desired file path.

Data pre-processing:

The data was manipulated for presentation using *importandprocessfunc*. A file containing the pixel-to-wavenumber mapping is additionally loaded and returned to the environment as *pixelwavenumbermap*. The spectral and temperature data is loaded and averaged based on time blocks specified by *resamplerate* (10 in this work). The total time must be divisible by the resample rate. Then the *spectraldata* and *backgrounddata* is rescaled by their respective CCD row counts (6 and 219, respectively) and the approximate dark current noise is subtracted (604.57 counts). Finally, the rescaled background data is subtracted from the rescaled spectral data and returned as *spectraldatasubtracted*. A *timedataresample* variable is also created and returned.

Linear changepoints of integrated peak area

For linear changepoints in the integrated intensity area evolution of peaks a suitable time window must be selected which should encompass an approximately linear approach and the dramatic slope change at the phase transition. This is necessary as the whole heating and cooling cycle may show many periodic trends, complicating the determination. It is useful to first roughly estimate the transition time by visualizing the entire heating process, such as in a surface plot. Then a smaller time window can be selected by specifying a custom range for *timeselect*.

For the changepoint analysis, first the desired wavenumber ranges to track are specified by their respective pixel ranges in *pix*, which may require some adjustments for clear results. *time* specifies the time range to analyze and should be iteratively determined until two linear sections are clearly achieved. The total integrated intensity areas of peaks is normalized from 0 to 1 for ease of visualization.

Selection of the pixel ranges which show a clear transition can now be specified in *pixelstep*. New *time1* and *time2* variables now specifies time ranges for the peaks that disappear and those that appear. The ideal new time ranges should encompass only the two linear trends arising from the actual Raman peaks of the phase being monitored. An additional variable *peakend* specifies whether a peak disappears (1) or appears (0) for the proper transition time sorting. The native MATLAB function *findchangepts* performs the linear change point analysis.

Finally, the transition start (*transitiontime1),* midpoint (*transitionmiddle*), end (*transitiontime2),* and *(transitiontotal)* are outputted.

The following is a list of the software installations needed:

MATLAB (Mathworks, Release R2021B)
    Data acquisition toolbox
LightField (Princeton Instruments)
    Add-ons and Automation SDK
GRBL (https://github.com/grbl/grbl)
Universal G-code Sender (https://winder.github.io/ugs_website/)

The following is a list of the MATLAB scripts included for instrument control and data analysis which are available to download from the location specified in our data availability statement. Appropriate file paths must be added before use.

startexperiment:

Specifies:
    acquiretime – total time for each individual spectrum acquisition
    numberacquires – number of spectra to acquire
Runs:
    startexperimentfunc
Procedures:
    Saves spectral and temperature data to .csv files

startexperimentfunc:

Inputs:

acquiretime

numberacquires

Runs:

Lightfield environment

thermocouplefunc

rasterfunc

Procedures:

Bins background spectra, reformats spectral data and background data

Calculates time offset between temperature data and spectral data

Outputs:

spectraldatareformat – spectral data in an array formatted as rows = pixels, columns = time

backgrounddatareformat – binned background data in an array formatted as rows = pixels, columns = time

thermocouplerecord – temperature data in array formatted as column 1 = time; column 2 = temperature in degrees Celsius

temperaturetospectraltimeoffset – time delay between temperature and spectral data (negative values indicate temperature is delayed relative to spectral data)

thermocouplefunc:

Inputs:

acquiretime – total time for each individual spectrum acquisition

numberacquires – number of spectra to acquire

Procedures:

Initialize connection to data acquisition device

Record time when temperature data recording begins

Record temperature sequentially at time intervals specified by *acquiretime* for *numberacquires* times

Outputs:

thermocouplerecord – array of temperatures in degrees Celsius

rasterfunc:

Inputs:

acquiretime – total time for each individual spectrum acquisition

numberacquires – number of spectra to acquire

Procedures:

Establish serial connection to rastering microcontroller

Specify dimensions of raster pattern *innerradius* and *outerradius*

Specify speed to move probe in mm/s *scanspeed*

Specify time to complete each movement *pausetime*

Send command to move probe into raster pattern

Send sequential commands to move probe in raster pattern until total time (*acquiretime*numberacquires)* is exceeded

Send command to return probe to central storage location

importandprocessfunc:

Inputs:

samplename – name associated with experiment data
centerwavelength – wavelength in nanometers that spectrometer grating is centered at to determine pixel to wavenumber mapping file
resamplerate – time window to average (*numberacquires* must be divisible by this number)

Procedures:

Import spectral data, background data, and temperature data associated with *samplename*
Import pixel to wavenumber mapping to create *pixelwavenumbermap*
Average the spectra and temperature in the time window specified by *resamplerate*
Create a *timedataresample* variable
Rescale spectral data and background data by their respective number of pixels
Subtract the approximate dark current of the instrument
Create *spectraldatasubtracted* by subtracting *backgrounddata* from *spectraldata*

Outputs:

spectraldatasubtracted
spectraldata
backgrounddata
pixelwavenumbermap
temperaturedataresample
timedataresample

transitionchangepointexample

Specifies:

samplename – name associated with experiment to analyze
centerwavelength
resample rate

Procedures:

Runs *importandprocessfunc*
Creates a full-time surface plot to estimate the transition time
Creates a limited-time surface plot to check transition (specified by *timeselect*)
Creates a plot of integrated intensity evolution of pixel ranges specified by *pix*
Determines the linear changepoints using *findchangepts*

Outputs:

initiation (*transitiontime1*), midpoint (*transitionmiddle*), end (*transitiontime2*), and total time (*transitiontotal)* based on the specification of pixel ranges (*peakend*) corresponding to Raman signals disappearing (peakend(n) = 1) or appearing (peakend(n) = 0).