Ask

* **Spike**: D3 support for Event markers and Setpoints
* **Goals**:

**Event markers:**

- Support for rendering given svgs at points based on x and y values (svg as ts file)

- Support for mouse interaction (hover and click)

- Support for showing tooltip on hover (text/html/angular component)

- Double click/ right click if component is not achieved

- Support for different svgs in the same row

**Setpoints:**

- Support for rendering of both vertical and horizontal lines on plot (setpoints and AFF).

- Support for rendering the lines with triangles at the end (same as figma screen)

- On Hover of setpoint lines tooltip should be shown (only text or html and angular component possible?)

- Advance and edge cases of multiple setpoints in continuation lines (standard to state based)

- Multiple setpoints overlayed (interaction and behaviour)

Approach

Event markers:

* **Support for rendering given svgs at points based on x and y values and Support for different svgs in the same row**

We can render the set of different svgs by providing x and y attributes along with ‘xlink:href’ attribute for rendering respective svg at given x and y values.

*private markers = [*

*{*

*x\_value: 120.469,*

*y\_value: 0.05,*

*img: '../../assets/arrow.svg',*

*tooltip\_text: 'arrow',*

*},*

*{*

*x\_value: 180.469,*

*y\_value: 0.05,*

*img: '../../assets/access-point.svg',*

*tooltip\_text: 'access point',*

*},*

*{*

*x\_value: 269.469,*

*y\_value: 0.05,*

*img: '../../assets/access-point-check.svg',*

*tooltip\_text: 'access point check',*

*},*

*{*

*x\_value: 320.469,*

*y\_value: 0.05,*

*img: '../../assets/access-point-off.svg',*

*tooltip\_text: 'access point off',*

*},*

*];*

*this.svg*

*.selectAll('.line')*

*.data(this.markers)*

*.enter()*

*.append('image')*

*.attr('x', (d: any) => {*

*return d.x\_value;*

*})*

*.attr('y', (d: any) => {*

*return d.y\_value;*

*})*

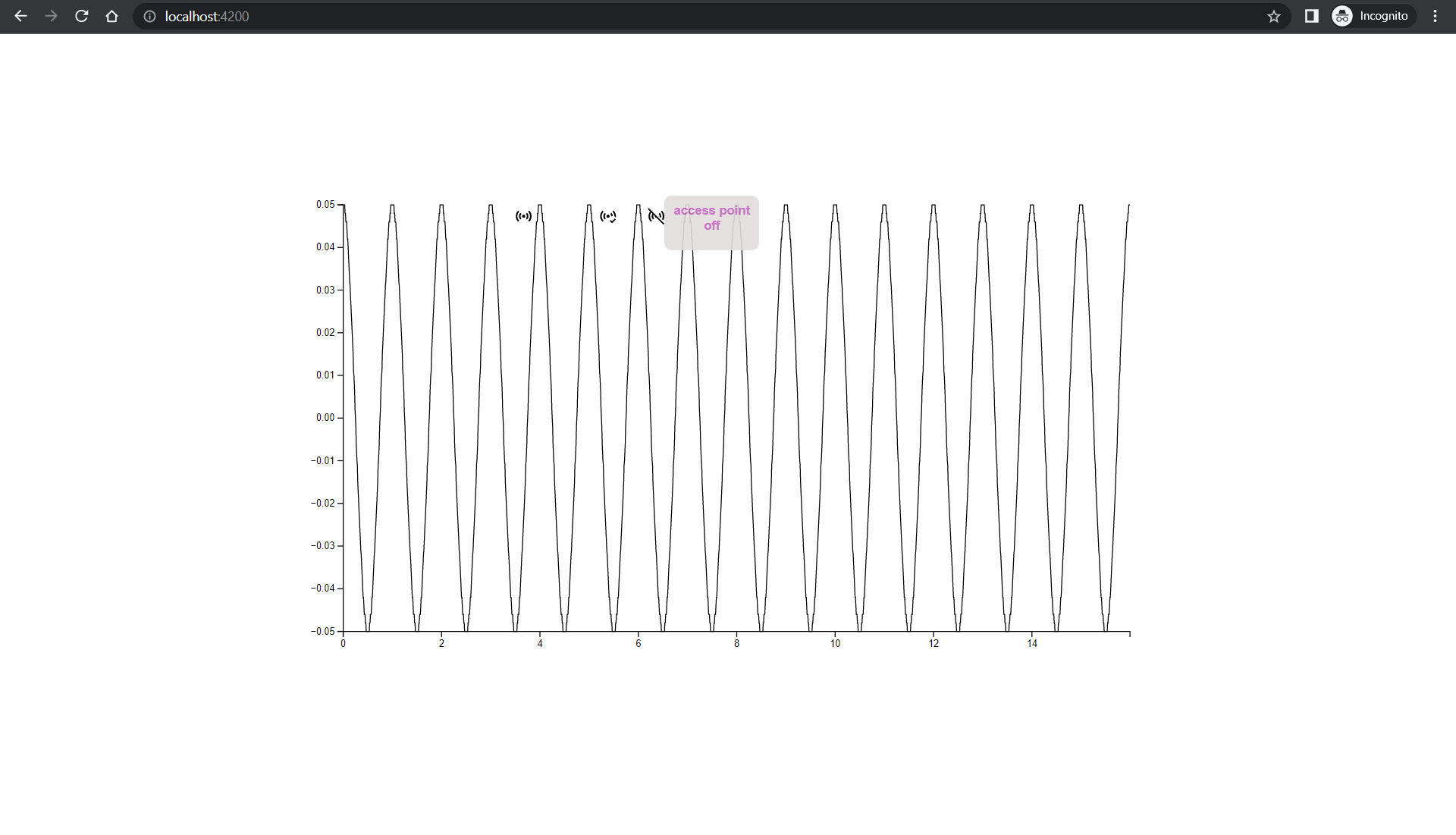
*.attr('width', 20)*

*.attr('height', 24)*

*.attr('xlink:href', (d: any) => {*

*return d.img;*

*})*



* **Support for mouse interaction (hover and click) and Support for showing tooltip on hover (text/html/angular component)**

We can handle the mouse interactions hover and click by providing respective event name and action to be performed accordingly. We can append a div as tooltip on mouse interaction where it facilitates the rendering of plain text/ html.

*.on('mouseover', (d: any) => {*

*this.tooltip.transition().duration(200).style('opacity', 0.9);*

*this.tooltip*

*.html(*

*'<h5 style="margin: 0px;">' +*

*d.target.\_\_data\_\_.tooltip\_text +*

*'</h5>'*

*)*

*.style('left', d.pageX + 'px')*

*.style('top', d.pageY - 28 + 'px');*

*})*

*.on('mouseout', () => {*

*this.tooltip.transition().duration(500).style('opacity', 0);*

*})*

*.on('click', this.click.bind(this))*

For rendering angular component, we can use a boolean and based on boolean value we can render angular component outside svg container

*this.tooltip*

*.html('<button id="button1">Test</button>')*

*.style('left', d.pageX + 'px')*

*.style('top', d.pageY - 28 + 'px')*

*.style('opacity', 1);*

*d3Selection*

*.select('#button1')*

*.on('click', () => {*

*this.selectClicked = true;*

*this.displaySelect.emit(this.selectClicked);*

*})*

*<div \*ngIf="displaySelect">*

*<h4>Select</h4>*

*<mat-form-field appearance="fill">*

*<select matNativeControl required>*

*<option value="volvo">Volvo</option>*

*<option value="saab">Saab</option>*

*<option value="mercedes">Mercedes</option>*

*<option value="audi">Audi</option>*

*</select>*

*</mat-form-field>*

*</div>*



* **Double click/ right click if component is not achieved**

Additional events like double click and right click can be handle by providing event names as ‘dblclick’ and ‘contextmenu’ respectively.

*.on('dblclick', (d: any) => {*

*alert('Double clicked');*

*})*

*.on('contextmenu', (d: any) => {*

*d.preventDefault(); //stops rendering default browser options on right click*

*this.tooltip.transition().duration(200).style('opacity', 0.9);*

*this.tooltip*

*.html(*

*'<button id="button1">Test</button>'*

*)*

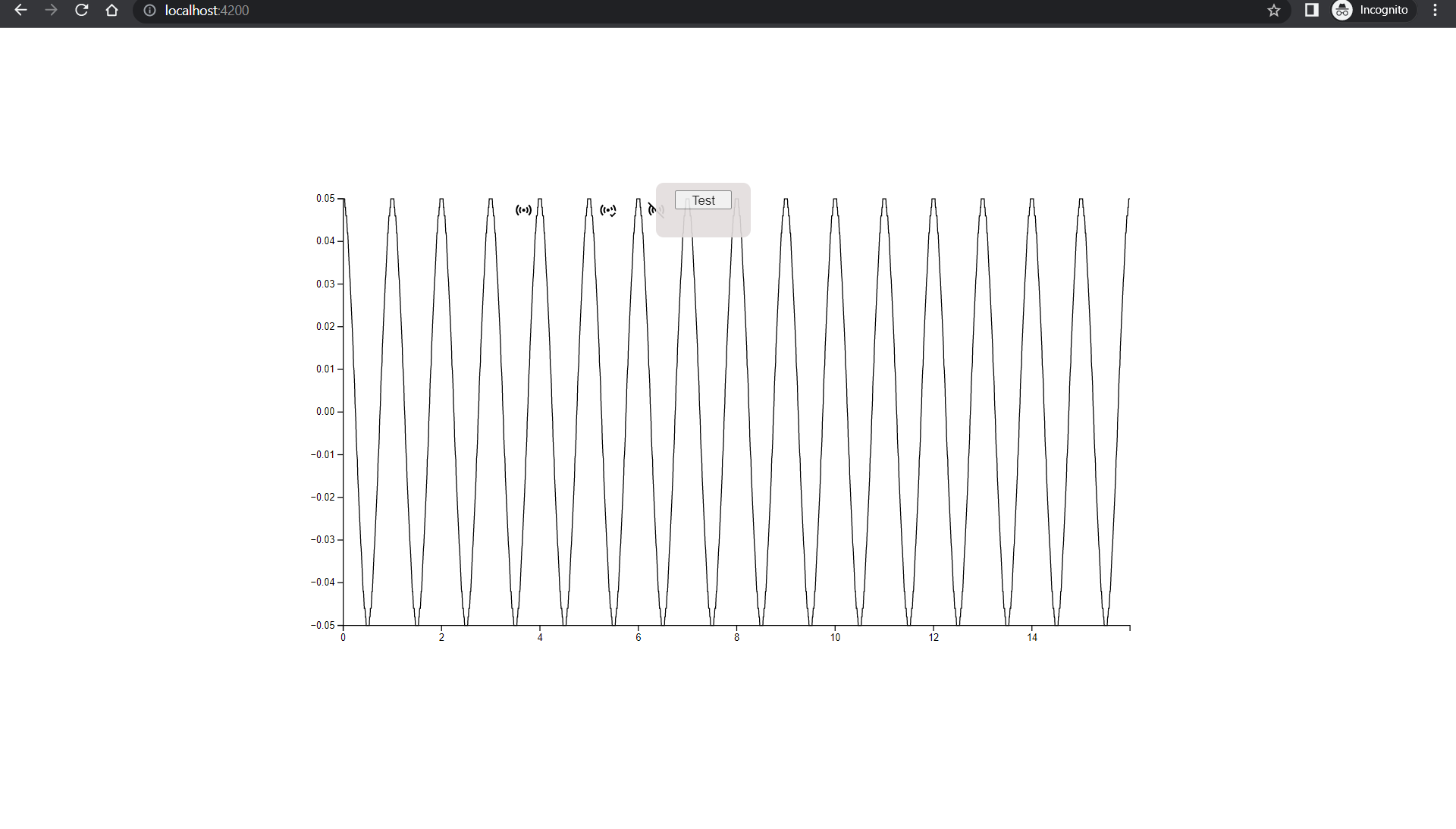
*.style('left', d.pageX + 'px')*

*.style('top', d.pageY - 28 + 'px');*

*d3Selection.select("#button1").on("click", ()=>{*

*alert("button 1 clicked")});*

*})*



Setpoints:

* **Support for rendering of both vertical and horizontal lines on plot (setpoints and AFF).**

Rendering of vertical lines for AFF and dashed horizontal lines for setpoints can be drawn by providing data with x and y values. Here for horizontal lines the x2 coordinate is considered as the plot width and for vertical lines y2-coordinate is considered as the plot height.

Vertical:

*this.svg*

*.selectAll('.line')*

*.data(this.setpoints)*

*.enter()*

*.append('line')*

*.attr('x1', (d: any) => {*

*return d.x\_value + 20;*

*})*

*.attr('y1', (d: any) => {*

*return d.y\_value;*

*})*

*.attr('x2', (d: any) => {*

*return d.x\_value + 20;*

*})*

*.attr('y2', this.height)*

Horizontal:

*this.svg*

*.selectAll('.line')*

*.data(this.setpointsDashed)*

*.enter()*

*.append('line')*

*.attr('x1', (d: any) => {*

*return d.x\_value;*

*})*

*.attr('y1', (d: any) => {*

*return d.y\_value;*

*})*

*.attr('x2', (d: any) => {*

*return this.width;*

*})*

*.attr('y2', (d: any) => {*

*return d.y\_value;*

*})*

*.style('stroke-width', 2)*

*.style('stroke', (d: any) => {*

*return d.color;*

*})*

*.style('stroke-dasharray', '3, 3');*

* **Support for rendering the lines with triangles at the end (same as figma screen)**

For adding triangle shape at both the ends of the lines we can use transform translate attribute with ‘d3Shape.symbolTriangle’. For downward triangles we can provide the rotate property in translate transform to adjust the direction of triangle.

*this.svg*

*.selectAll('.line')*

*.data(this.setpointsDashed)*

*.enter()*

*.append('path')*

*.attr('d', d3.symbol().type(d3Shape.symbolTriangle))*

*.attr('transform', (d: any) => {*

*if (d.direction == 'up')*

*return 'translate(' + (d.x\_value + 5) + ',' + (d.y\_value - 2.5) + ')';*

*return (*

*'translate(' +*

*(d.x\_value + 5) +*

*',' +*

*(d.y\_value + 2.5) +*

*') rotate(180)'*

*);*

*})*

* **On Hover of setpoint lines tooltip should be shown (only text or html and angular component possible?)**

For hover functionality on setpoint lines, the tooltip features which are applicable for events markers can also be achieved on setpoint lines

* **Multiple setpoints overlayed (interaction and behaviour)**

Multiple setpoints with different levels can be overlayed and overlapping tooltips can be handled with the required logic

*.on('mouseover', (d: any) => {*

*this.tooltip.transition().duration(200).style('opacity', 0.9);*

*let txt;*

*let count = 0;*

*for (let i = 1; i < this.setpointsDashed.length; i++) {*

*if (d.target.\_\_data\_\_.y\_value == this.setpointsDashed[i].y\_value) {*

*if (*

*count == 0 &&*

*d.target.\_\_data\_\_.tooltip !== this.setpointsDashed[i].tooltip*

*)*

*txt =*

*d.target.\_\_data\_\_.tooltip +*

*'<br/>' +*

*this.setpointsDashed[i].tooltip;*

*if (*

*count == 0 &&*

*d.target.\_\_data\_\_.tooltip === this.setpointsDashed[i].tooltip*

*)*

*txt = d.target.\_\_data\_\_.tooltip;*

*count++;*

*} else txt = d.target.\_\_data\_\_.tooltip;*

*}*

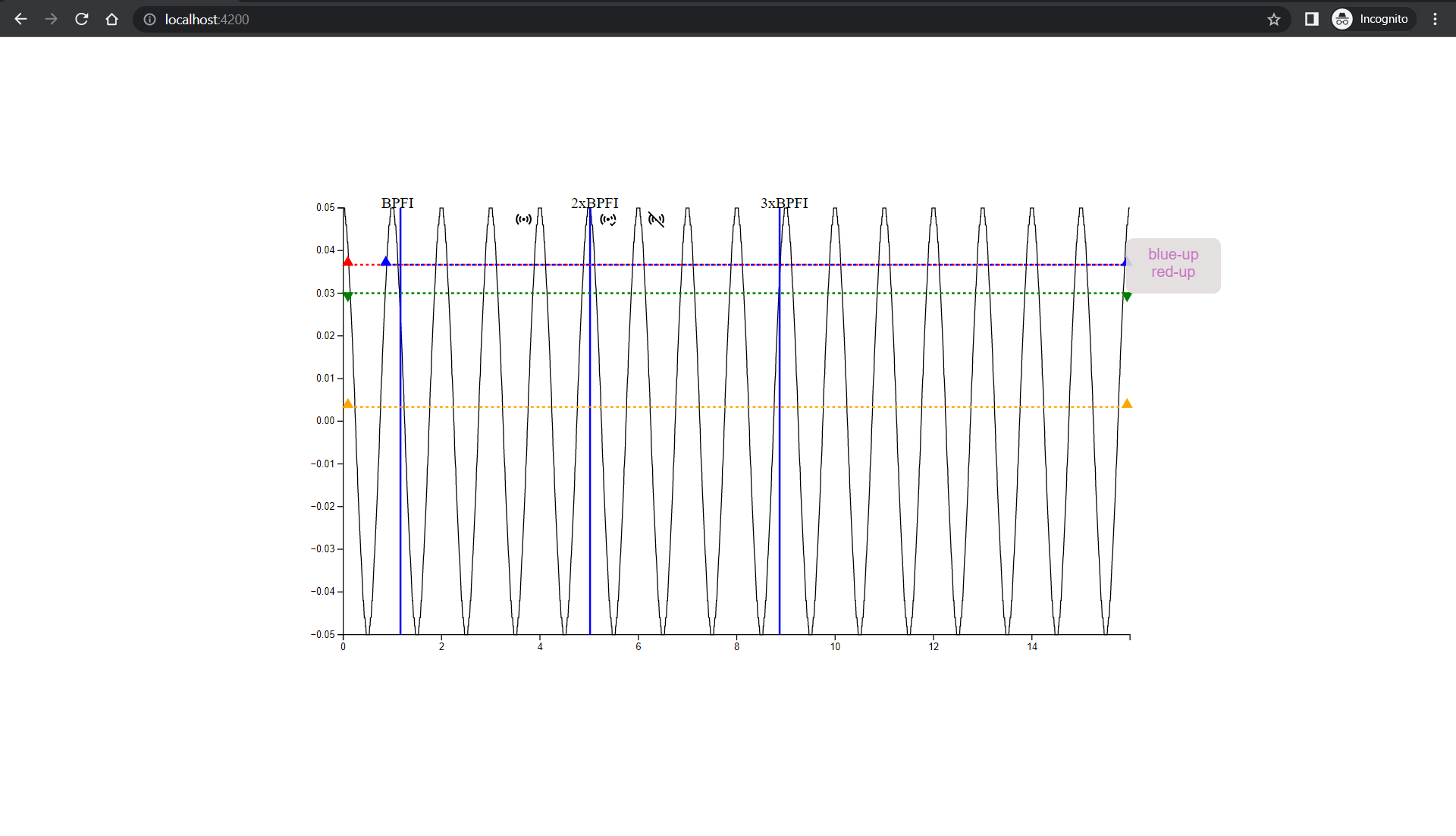
*this.tooltip*

*.html(txt)*

*.style('left', d.pageX + 'px')*

*.style('top', d.pageY - 28 + 'px');*

*})*



Conclusion

* From above observations we can conclude thatdD3 library supports rendering different svgs(considered as event markers) in single row w.r.t same plot and provides functionality for handling different events on the svgs.
* D3 also supports rendering horizontal and vertical lines (considered as setpoints and AFF) and handling respective events on shapes over setpoint lines.
* Future use cases which have to be implemented are
  + Functioning of event markers and setpoints on zoom
  + Event marker as subplot vs same plot.
  + Advance and edge cases of multiple setpoints in continuation lines

Resources

* <https://github.com/likhitha-pakanati /d3-plots-spike/tree/event-markers>
* <https://github.com/d3/d3/blob/main/API.md>