Short tutorial on TikZ

Drawing with exact precision

Pritam Karmokar

[Weekly Meeting 03/12/2021]

UNIVERSITY OF TEXAS
ROBOTIC
VISION

Table of Contents

Table of Contents

What is TikZ?

TikZ...

What is TikZ?

TikZ...

- ...is a recursive acronym for "TikZ *ist kein Zeichenprogramm*"

What is TikZ?

TikZ...

- ...is a recursive acronym for "TikZ *ist kein Zeichenprogramm*"
- is probably the most complex and powerful tool for creating graphic elements in LaTeX

What is TikZ?

TikZ...

- ...is a recursive acronym for "TikZ *ist kein Zeichenprogramm*"
- is probably the most complex and powerful tool for creating graphic elements in LaTeX
- defines a number of TeX commands that draw graphics

Good and Bad ☯

Pros:

- Quick creation of simple graphics

Good and Bad ☯

Pros:

- Quick creation of simple graphics
- Drawing programmatically with exact precision

Good and Bad ☯

Pros:

- Quick creation of simple graphics
- Drawing programmatically with exact precision
- Superior consistent typography

Good and Bad ☯

Pros:

- Quick creation of simple graphics
- Drawing programmatically with exact precision
- Superior consistent typography

Cons:

- Steep learning curve

Good and Bad ☯

Pros:

- Quick creation of simple graphics
- Drawing programmatically with exact precision
- Superior consistent typography

Cons:

- Steep learning curve
- No WYSIWYG

Good and Bad ☯

Pros:

- Quick creation of simple graphics
- Drawing programmatically with exact precision
- Superior consistent typography

Cons:

- Steep learning curve
- No WYSIWYG
- Changes require recompilation

Quick examples

Example (line)

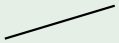
```
\tikz \draw (0pt,0pt) -- (20pt,6pt);
```

yields

Quick examples

Example (line)

```
\tikz \draw (0pt,0pt) -- (20pt,6pt);
```

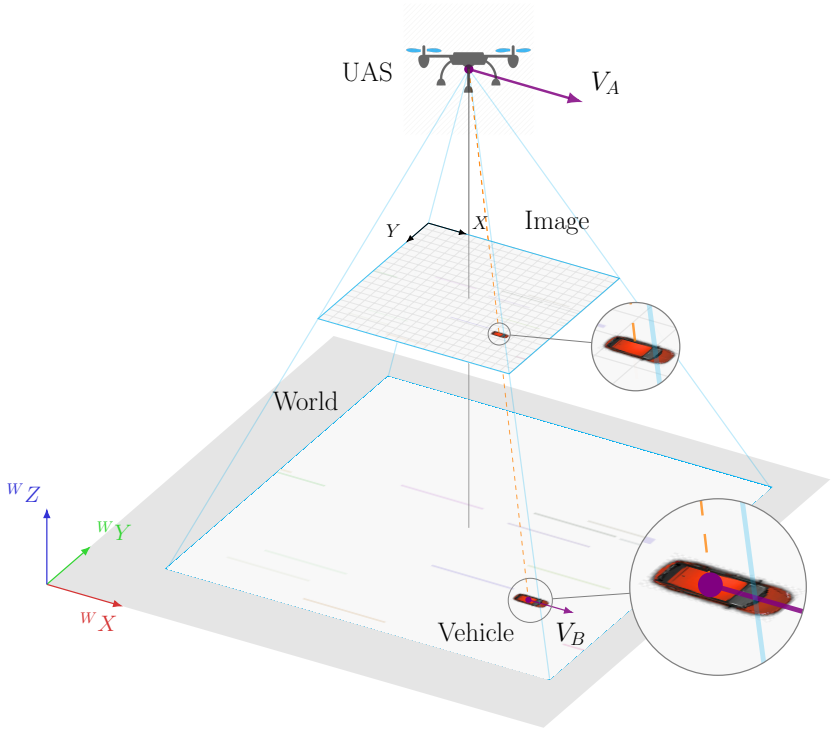
yields 

Example (orange)

```
\tikz \fill[orange] (1ex,1ex) circle (1ex);
```

yields 

Practical examples



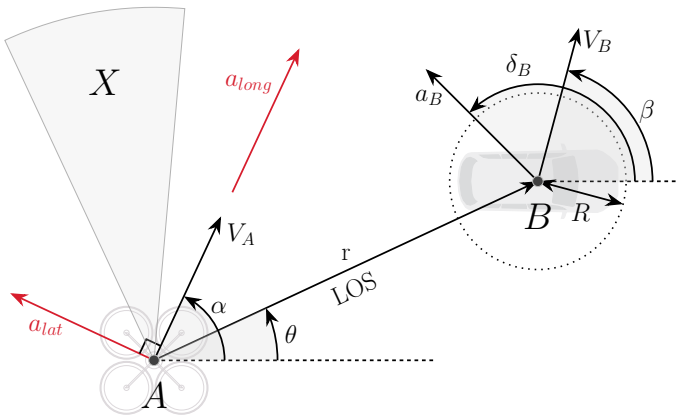


Table of Contents

Hello, TikZ!

```
\begin{tikzpicture}⟨animations spec⟩[⟨options⟩]
    ⟨environment contents⟩
\end{tikzpicture}
```

Hello, TikZ!

```
\begin{tikzpicture}⟨animations spec⟩[⟨options⟩]
    ⟨environment contents⟩
\end{tikzpicture}
```

Example (tikzpicture)

```
\usepackage{tikzpicture}  %preamble
...
\begin{document}
    \begin{tikzpicture}
        \path[draw] (0pt,0pt) -- (20pt,6pt);
    \end{tikzpicture}
\end{document}
```

Table of Contents

Basic Design Principles

1. Special syntax for specifying points
2. Special syntax for path specifications

Basic Design Principles

1. Special syntax for specifying points
2. Special syntax for path specifications
3. Actions on paths

Basic Design Principles

1. Special syntax for specifying points
2. Special syntax for path specifications
3. Actions on paths
4. Key-value syntax for graphic parameters

Basic Design Principles

1. Special syntax for specifying points
2. Special syntax for path specifications
3. Actions on paths
4. Key-value syntax for graphic parameters
5. Special syntax for nodes
6. Special syntax for trees
7. Special syntax for graphs

Basic Design Principles

1. Special syntax for specifying points
2. Special syntax for path specifications
3. Actions on paths
4. Key-value syntax for graphic parameters
5. Special syntax for nodes
6. Special syntax for trees
7. Special syntax for graphs
8. Grouping of graphic parameters

Basic Design Principles

1. Special syntax for specifying points
2. Special syntax for path specifications
3. Actions on paths
4. Key-value syntax for graphic parameters
5. Special syntax for nodes
6. Special syntax for trees
7. Special syntax for graphs
8. Grouping of graphic parameters
9. Coordinate transformation system

Special syntax for specifying points

- Cartesian: `(1cm,2pt)` or Polar: `(30:1cm)`
- Default units: length - centimeters (cm), angle - degrees (°)

Special syntax for specifying points

- Cartesian: `(1cm,2pt)` or Polar: `(30:1cm)`
- Default units: length - centimeters (cm), angle - degrees (°)
- 3D xyz-coordinate: `(1,2,3)`

Special syntax for specifying points

- Cartesian: `(1cm,2pt)` or Polar: `(30:1cm)`
- Default units: length - centimeters (cm), angle - degrees (°)
- 3D xyz-coordinate: `(1,2,3)`
- Anchor as coordinate: `(mynode_1 node.south)`

Special syntax for specifying points

- Cartesian: `(1cm,2pt)` or Polar: `(30:1cm)`
- Default units: length - centimeters (cm), angle - degrees (°)
- 3D xyz-coordinate: `(1,2,3)`
- Anchor as coordinate: `(mynode_1 node.south)`
- Relative coordinates *(1) change current point (2) don't change*
  1. `(1,0)`, `++(1,0)`, `++(0,1)` would specify `(1,0)`, then `(2,0)`, and `(2,1)`
  2. `(1,0)`, `+(1,0)`, `+(0,1)` would specify `(1,0)`, then `(2,0)`, and `(1,1)`

Special syntax for path specifications

- A *path* is a series of straight or curved lines, which need not be connected

Special syntax for path specifications

- A *path* is a series of straight or curved lines, which need not be connected
- Path specification is the main part while creating TikZ pictures

Special syntax for path specifications

- A *path* is a series of straight or curved lines, which need not be connected
- Path specification is the main part while creating TikZ pictures

Example (*path* specification)

```
\tikz{\filldraw[fill=white,draw=red,]
        (5pt,0pt)--(0pt,0pt)--(0pt,5pt)--cycle;}
```

yields

Actions on paths

- A *path* is a series of straight or curved lines, which need not be connected

Actions on paths

- A *path* is a series of straight or curved lines, which need not be connected
- We need to specify what needs to happen with it

Actions on paths

- A *path* is a series of straight or curved lines, which need not be connected
- We need to specify what needs to happen with it
- *draw*, *fill*, *shade*, *clip* or a combination?

Actions on paths

- A *path* is a series of straight or curved lines, which need not be connected
- We need to specify what needs to happen with it
- *draw*, *fill*, *shade*, *clip* or a combination?
- `\draw` is shorthand for `\path[draw]`

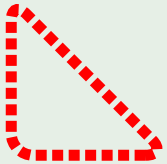
Actions on paths

- A *path* is a series of straight or curved lines, which need not be connected
- We need to specify what needs to happen with it
- *draw*, *fill*, *shade*, *clip* or a combination?
- `\draw` is shorthand for `\path[draw]`
- Likewise, `\fill`, `\shade`, `\clip`

Key-value syntax for graphic parameters

Example (Key-value)

```
\tikz{
    \draw[
            line width=2pt,color=red,
            densely dotted,rounded corners,
        ]
            (1,0) -- (0,0) -- (0,1) -- cycle;
    }
```

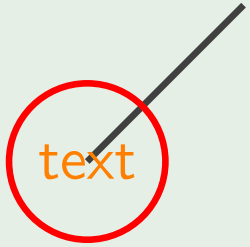
yields 

Special syntax for nodes

Example (nodes)

```
\tikz{
    \draw[darkgray,very thick]
    (1,1)
    node[circle,draw=red] (dummytextnode)
    {\textcolor{orange}{text}}
    -- (2,2);
}
```

yields

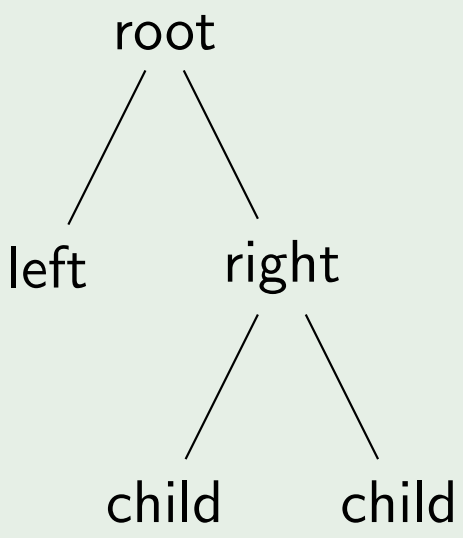


Special syntax for trees

Example (trees)

```
\begin{tikzpicture}
    \node {root}
    child {node {left}}
    child {node {right}
        child {node {child}}
        child {node {child}}
    };
\end{tikzpicture}
```

yields

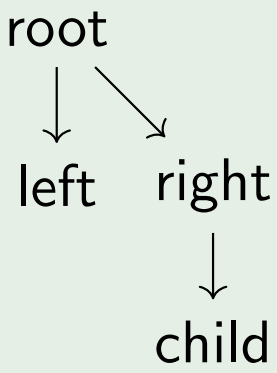


Special syntax for graphs

Example (graphs)

```
\usetikzlibrary {graphs}
\tikz
    \graph[
        grow down,
        branch right] {
    root->{left,
        right->{child,
            child}}};
```

yields



Grouping of graphic parameters

Example (grouping graphic parameters)

```
\begin{tikzpicture}
\begin{scope}[red]
\draw[ultra thick]
    (0mm,8mm)--(10mm,8mm);
\end{scope}
\begin{scope}[green]
\draw[ultra thick]
    (0mm,4mm)--(10mm,4mm);
\draw[blue,ultra thick]
    (0mm,0mm)--(10mm,0mm);
\end{scope}
\end{tikzpicture}
```

yields

Coordinate transformation system

Example (coordinate transformation)

```
\begin{tikzpicture}
    \node[ellipse,draw]
    at (0,0) {(0,0) before};
    \tikzset{yshift=10mm}
    \node[ellipse,draw]
    at (0,0) {(0,0) after};
\end{tikzpicture}
```

yields

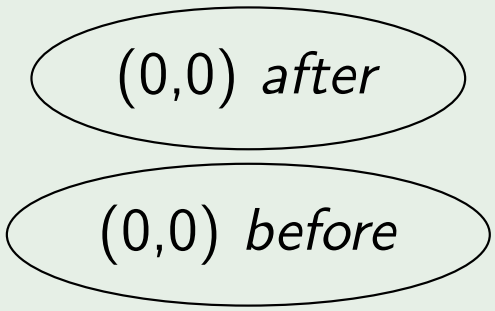


Table of Contents

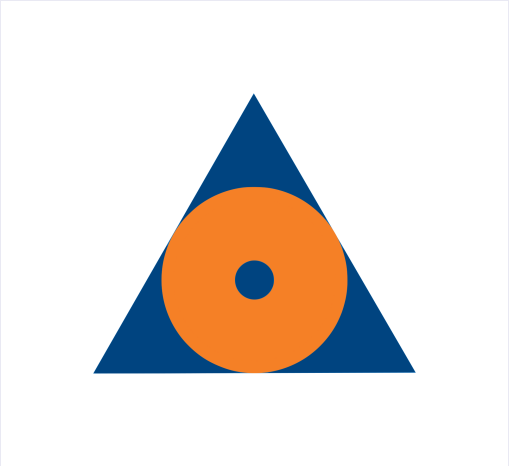
Getting acquainted

How could we draw this?

- A triangle, an incircle and an inner circle

Getting acquainted

Break down



- Triangle can be created by *filling* through it's three coords
- Circles can be created readily by *filling circles* at the *incenter* coordinate

Getting acquainted

Walk through

```
\coordinate (A) at (0,0);
\coordinate (B) at (0,1cm);
\coordinate (C) at (60:1cm);
\coordinate (I) at (5mm,{1*sqrt(3)/6});
```

Getting acquainted

Wrapping up

```
\definecolor{YALE_BLUE}{RGB}{0, 68, 128}
\definecolor{PRINCETON_ORANGE}{RGB}{245, 128, 38}

\filldraw[fill=YALE_BLUE,draw=none]
    (A) -- (B) -- (C) -- cycle;
\node[circle,minimum size={sqrt(3)/3},
    fill=PRINCETON_ORANGE,draw=none,] at (I) {};
\node[circle,minimum size={(1.25/pi)*sqrt(3)/6},
    fill=YALE_BLUE,draw=none,] at (I) {};
```

Getting acquainted

The outcome

Getting acquainted

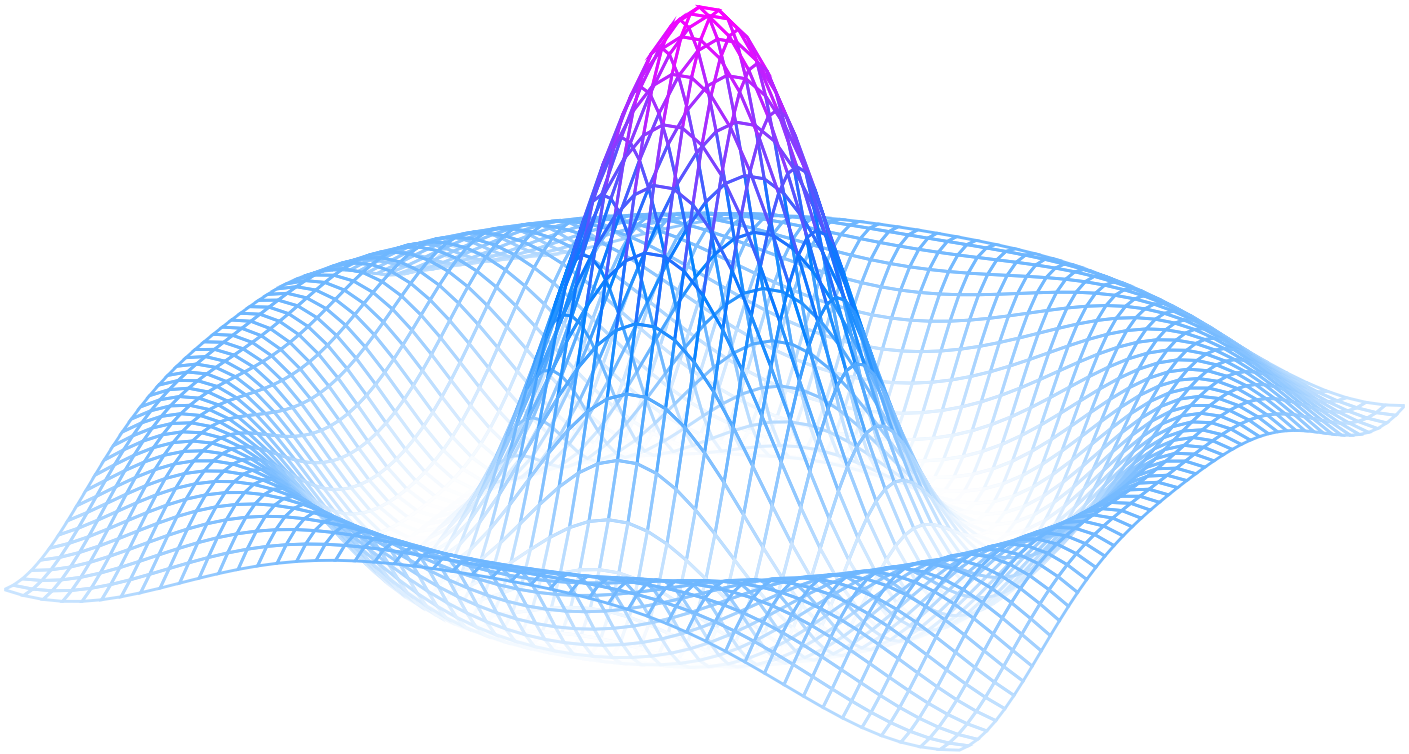
The outcome...

Table of Contents

Hello, PGFPLOTS!

Portable Graphics Format

$$\frac{sin(r)}{r}$$



Hello, PGFPLOTS!

- PGFPLOTS is built completely on TikZ/PGF

Hello, PGFPLOTS!

- PGFPLOTS is built completely on TikZ/PGF
- Knowledge of TikZ will simplify the work with PGFPLOTS

Hello, PGFPLOTS!

- PGFPLOTS is built completely on TikZ/PGF
- Knowledge of TikZ will simplify the work with PGFPLOTS
- PGFPLOTS comes with two components:

Hello, PGFPLOTS!

- PGFPLOTS is built completely on TikZ/PGF
- Knowledge of TikZ will simplify the work with PGFPLOTS
- PGFPLOTS comes with two components:
  1. the plotting component

Hello, PGFPLOTS!

- PGFPLOTS is built completely on TikZ/PGF
- Knowledge of TikZ will simplify the work with PGFPLOTS
- PGFPLOTS comes with two components:
  1. the plotting component
  2. PGFPLOTSTABLE component which simplifies number formatting and postprocessing of numerical tables. (separate package)

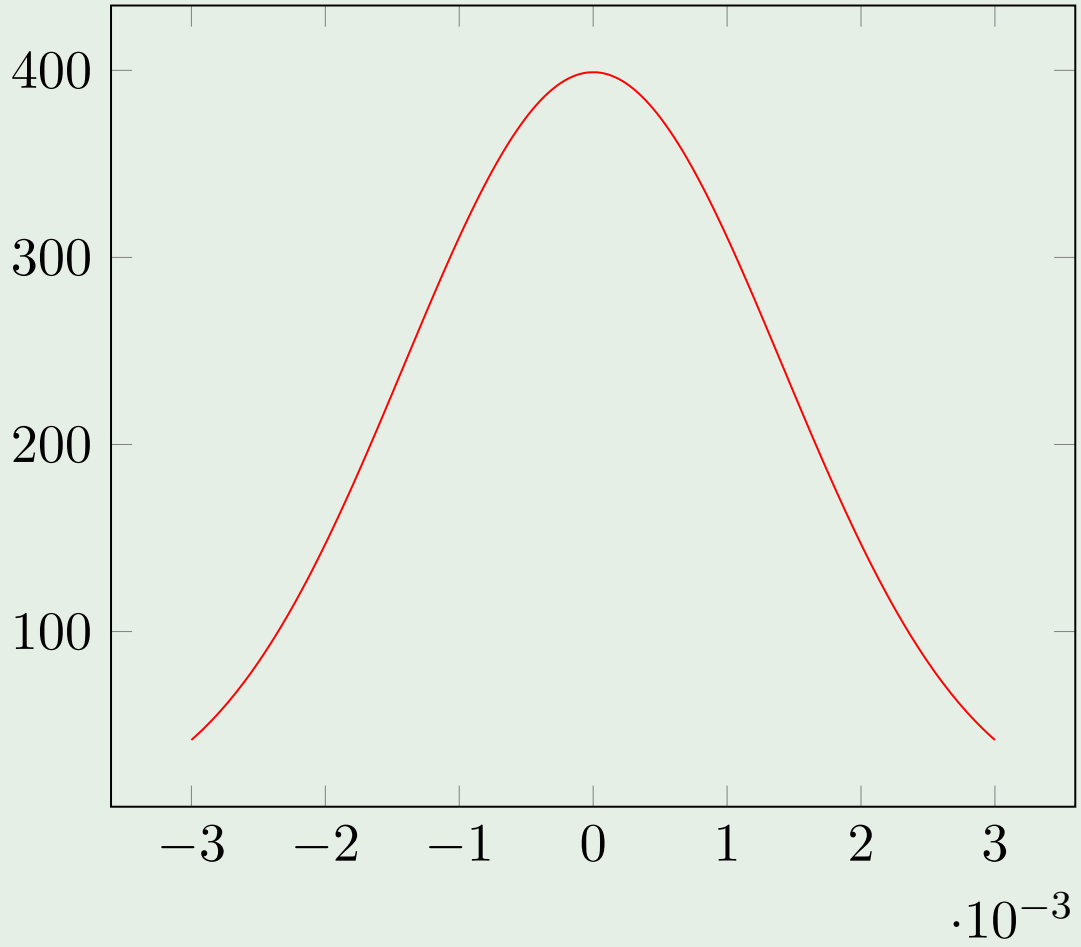
A quick intro

Example (code)

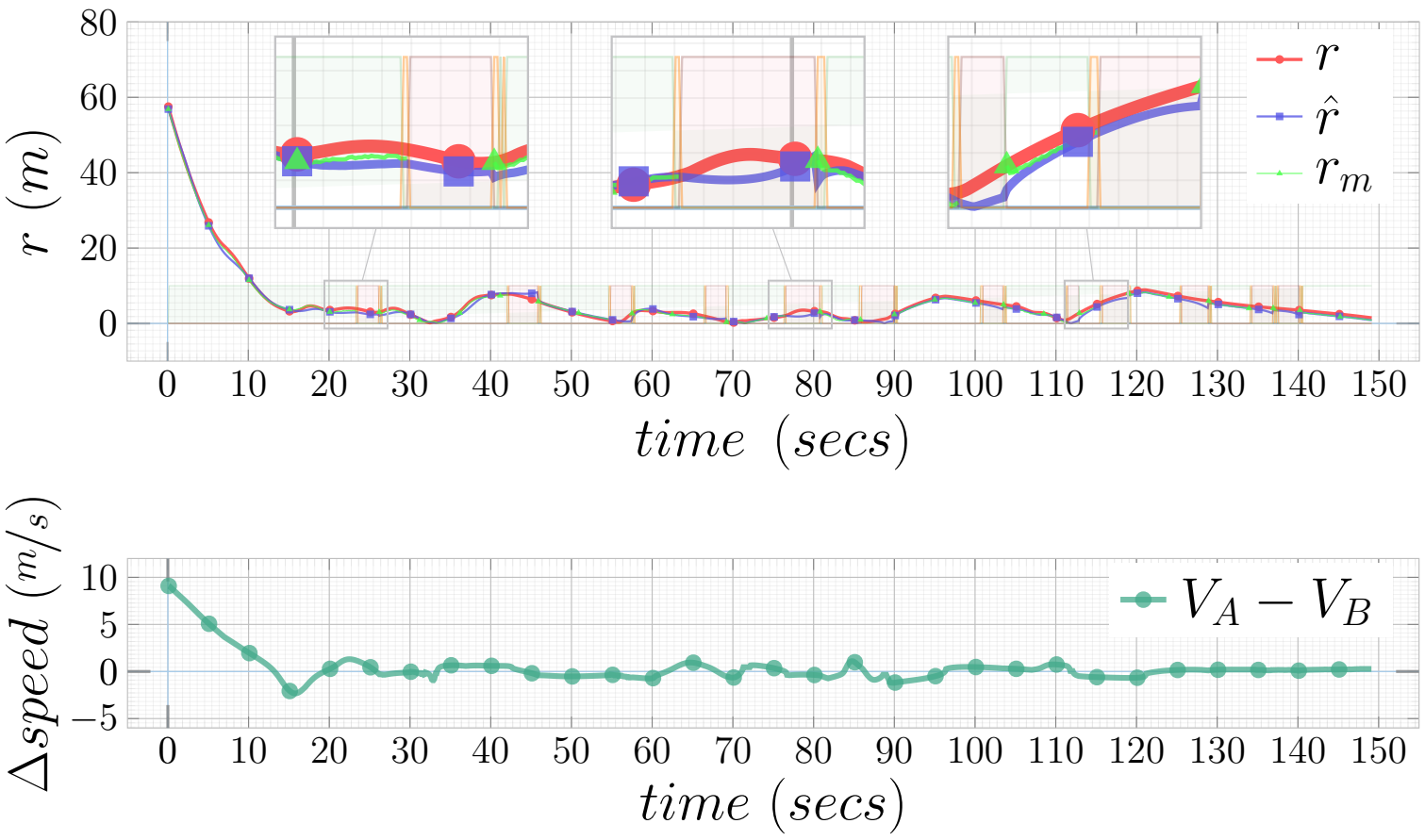
```
\begin{tikzpicture}
    \begin{axis}[]
        \addplot [
            red,
            domain=-3e-3:3e-3,
            samples=201,
        ]
        {exp(-x^2 / (2e-3^2)) / (1e-3 * sqrt(2*pi))};
    \end{axis}
\end{tikzpicture}
```

A quick intro

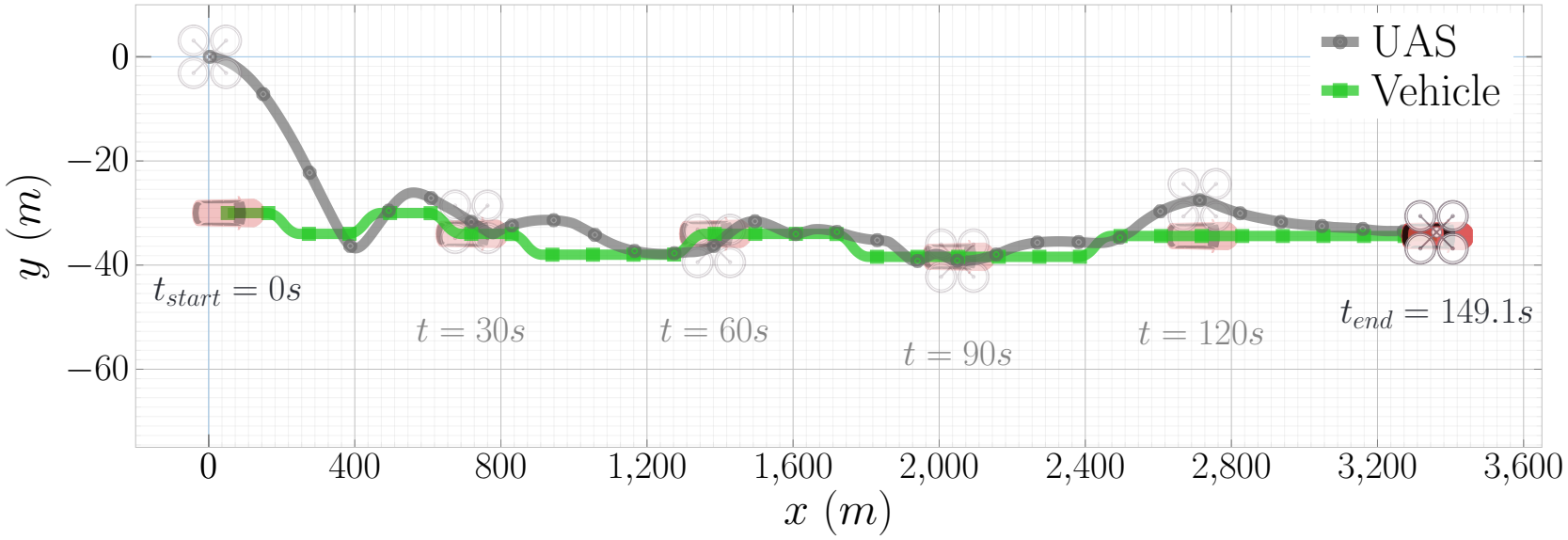
Example (output)



In practice


80
60
40
20
0
$r\ (m)$
r
$\hat{r}$
r_m
0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150
$time\ (secs)$
10
5
0
−5
$\Delta speed\ (m/s)$
$V_A - V_B$
0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150
$time\ (secs)$

In practice


UAS
Vehicle
0
−20
−40
−60
y (m)
$t_{start} = 0s$
$t = 30s$
$t = 60s$
$t = 90s$
$t = 120s$
$t_{end} = 149.1s$
0
400
800
1,200
1,600
2,000
2,400
2,800
3,200
3,600
x (m)

That's all folks!

Some useful resources

- Online TikZ/PGF web manual
- Overleaf TikZ doc
- RVL archives
- Scientific pictures show off
- Awesome TikZ repos

Table of Contents

To be continued..

Summary

- TikZ is cryptic, painful but an awesome drawing package
- Like some other skills it will need constant touch
- PGFPLOTS and PGFPLOTSTABLE can aid our process of making quality illustrative figures and tables in research papers

Thank you

Fin.