Robotics Masterclass

Harper Adams University

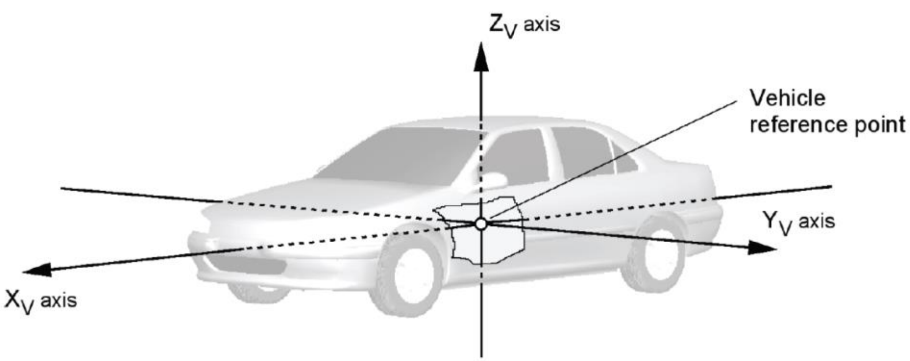
Robotics Masterclass

Harper Adams University

Week	Date	Lecture (1hr) Files	Date THURSDAYS	Student practical (1-2hr)
1	22 April MONDAY	Mobile robots, connecting, configuration, I/O, A/D	25 April THURSDAY	Recap, student demonstrations, Q&A
2	29 April MONDAY	Kinematics, CAN Communication 06-Kinematics.docx	2 May THURSDAY	Recap, student demonstrations, Q&A
3	8 May WEDNESDAY	Sensors and actuators	9 May THURSDAY	Recap, student demonstrations, Q&A
4	13 May MONDAY	Filters,Control and feedback, PID	16 May THURSDAY	Recap, student demonstrations, Q&A
5	20 May MONDAY	Navigation using GPS	23 May THURSDAY	Recap, student demonstrations, Q&A
holiday				
6	4 June TUESDAY	States and behaviour	7 June FRIDAY	Recap, student demonstrations, Q&A
7	10 June MONDAY	Path planning	13 June THURSDAY	Recap, student demonstrations, Q&A
holiday				
8	25 June TUESDAY	X track error, Communication using wireless, sending a character to control. Teleoperation, safety dead-man's handle	27 June THURSDAY	Recap, student demonstrations, Q&A
9	1 July MONDAY	Demonstrations	5 July FRIDAY	Recap, student demonstrations, Q&A

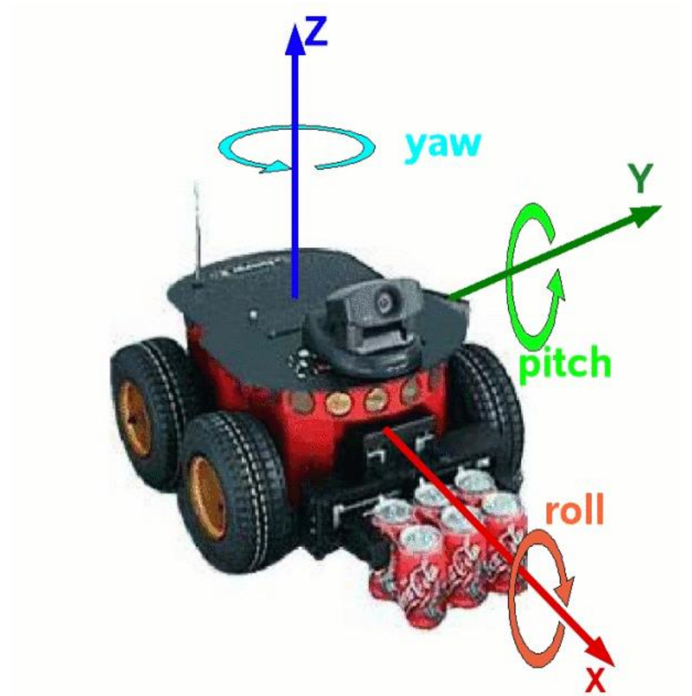
Shared drive: github.com/swane/tafe

Coordinate frame definition



A. VEHICLE AXIS SYSTEM – Z-UP

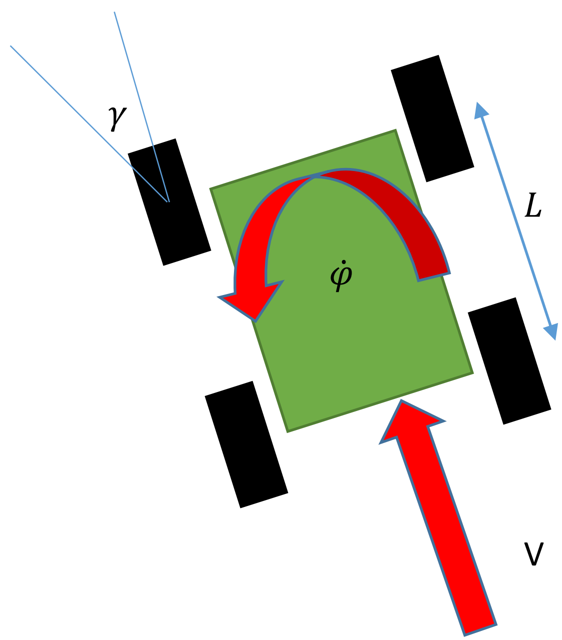
Society of Automotive Engineers (SAE) J670e (wiki.uncee.org)




OVER HERE
NO, THIS WAY

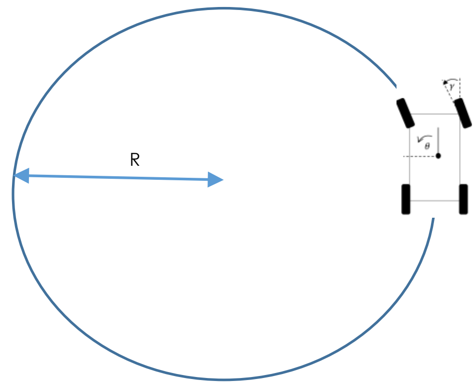
Rotation velocity related to steer angle

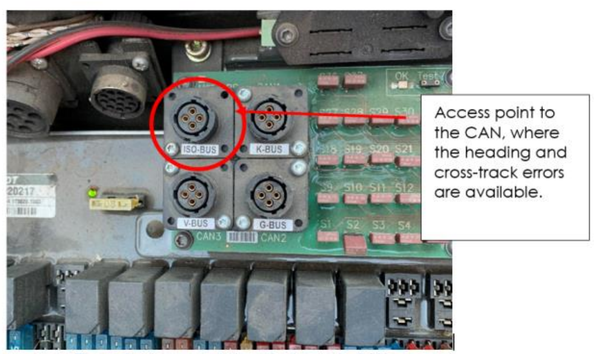
$$\dot{\varphi} = \frac{V \tan\gamma}{L}$$



Radius of turn related to steer angle

The radius of turn is related to the steer angle by: $R = \dfrac{L}{\tan(-\gamma)}$

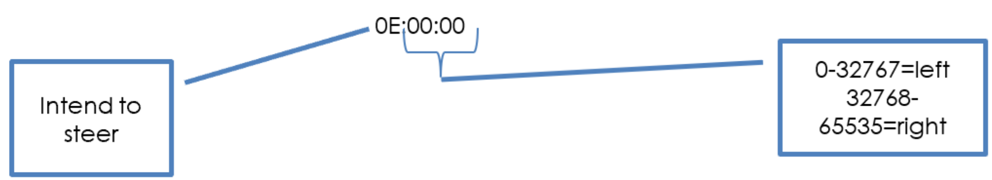




Harper Adams University

CAN Command

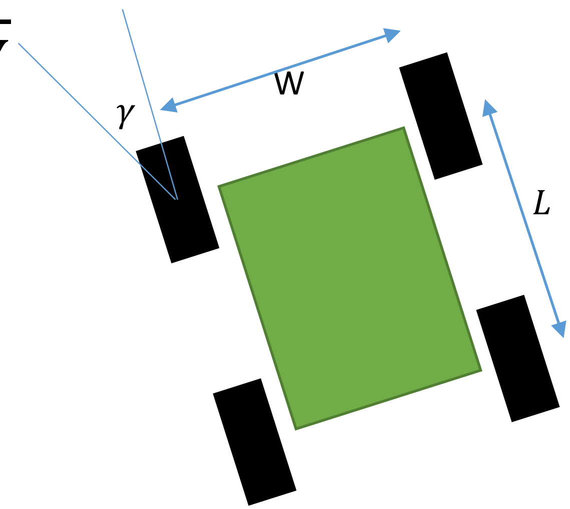
- The CAN bus can control the vehicle steering when it has ‘SteerCAN’ fitted. This is is 250kbit/s signal from the tractor electronic control unit (TECU). The TECU comprises of: wheel angle senor and a steer valve and it commanded to steer to a curvature (1/radius).
- The CAN data issues the command ‘Intend to Steer’ this is control code bytes representing



1/R

In the case of vehicle control, the steer angle is sent as the inverse of the radius (this stops the possibility of sending infinity to drive in a straight line). The steer angle is related to 1/R by:

$$^{1}/_{R} = \frac{\tan(-\gamma)}{L} + \frac{2}{W}$$



Steer commands

All Examples at: Examples→ATLAS→Steering

steer(double steer_angle);

Input: double steer angle

Returns: nothing

Function: sets the front steer angle in degrees with 0 – centre, and positive is steer right

steer_radius(double inv_R);

Input: double, inverse radius

Returns: The steer angle calculated from the vehicle length.

Function: returns the front steer angle to steer to a radius where a positive value will steer LEFT.

Note: inv_R = 1 / Radius;

set_veh_length(double v)

You can change the vehicle length as used by the 'steer_radius' calculations here. It is set to 0.6m by default.

set_veh_width(double v)

You can change the vehicle length as used by the 'steer_radius' calculations here. It is set to 0.5m by default.

rear_steer(double steer_angle);

Input: double steer angle

Returns: nothing

Function: sets the steer angle in degrees with 0 – centre, and positive is steer right

E.G. Steer to a radius of 2m:

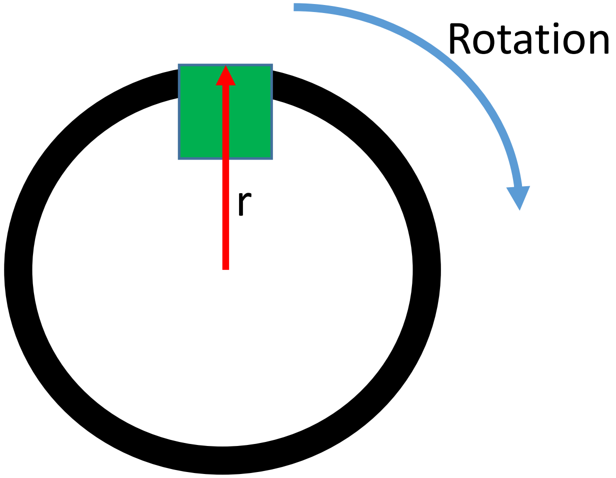
double Radius=2;

double inv_R = 1 / Radius;

steer(-steer_radius(inv_R));

How far ?

- We count the number of revolutions ‘n’

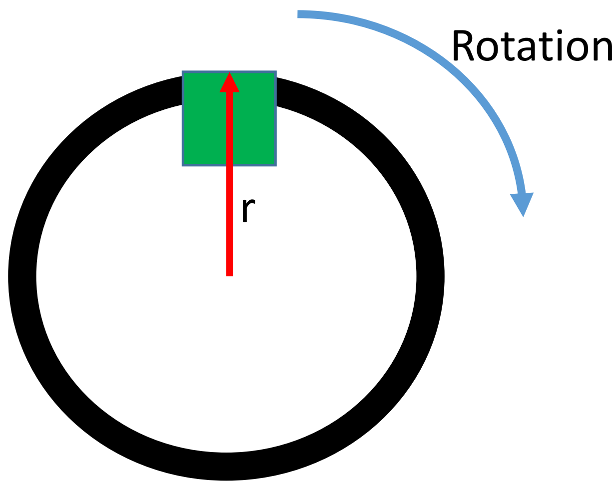


- Distance=$2\pi r$ x n, or circumference x n

Harper Adams University

How fast ?

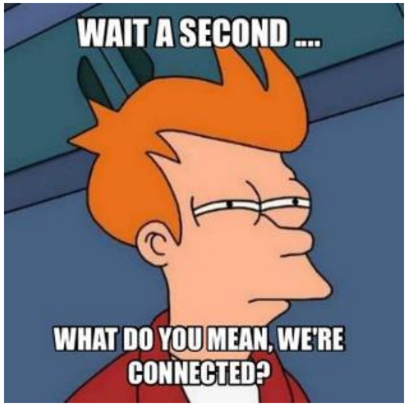
- We count the number of revolutions ‘n’ in a fixed time

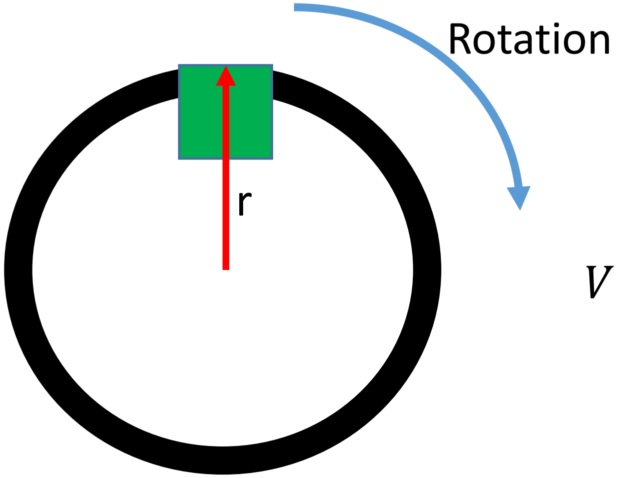


- $V = 2\pi r \times n$ / time (secs)

How fast ?

- Or the time between revolutions?





$$V \propto \frac{1}{time\ between\ pulses}$$

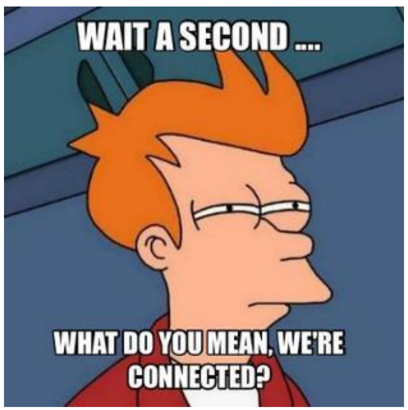
- V=2πr / time (secs)

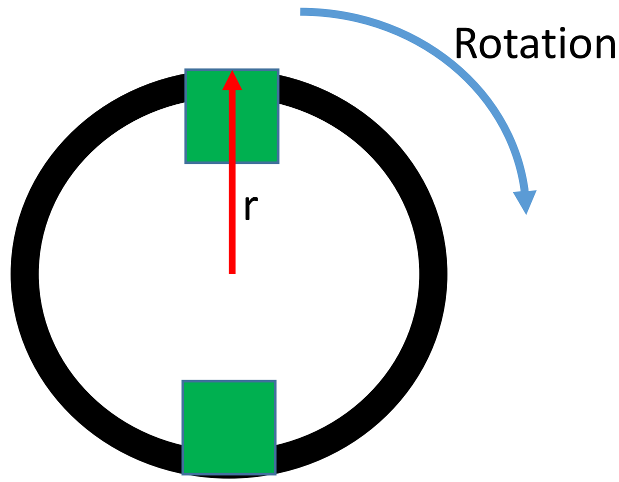
$$V = \frac{distance}{time}$$

Harper Adams University

Doubling the magnets

- Or the time between ½ revolutions?





- $V = 2\pi r / 2 / \text{time (secs)}$

Examples→wheel_speed

Harper Adams University

Pre-set values for wheel measure- uses the wheel encoder

- Pre-set values:
  - hall_ppr = 2; the number of magnets on the wheel
  - wheel_circ = 0.42; circumference in m

- To set your own values:
  - void set_hall_ppr(double v)
  - void set_wheel_circ(double v)

Harper Adams University

Reading the wheel encoder

- void zero_wheel()
- unsigned long read_wheel() (wheel pulses)
- unsigned long read_wheel_time() (microseconds)
- double veh_speed() (m/s)

```
{
    unsigned long wv;
    wv=read_wheel_time();
    double spd=wheel_circ / hall_ppr / ((double)wv*0.00001);
    return spd;
}
```

- double veh_dist(); (metres) coming in next update

Harper Adams University

Interrupt to count wheel encoder

This is an internal function called by an interrupt on the Hall wheel sensor:

```
void countWheel() //HALL SENSOR interrupt
{
  static unsigned long lastt;
  if (digitalRead(HALL_SENSOR)) {wheel++;
t_between_wheel_pulses=micros()-lastt;}
lastt=micros();
}
```

- You cannot access it as it is used by the system only.

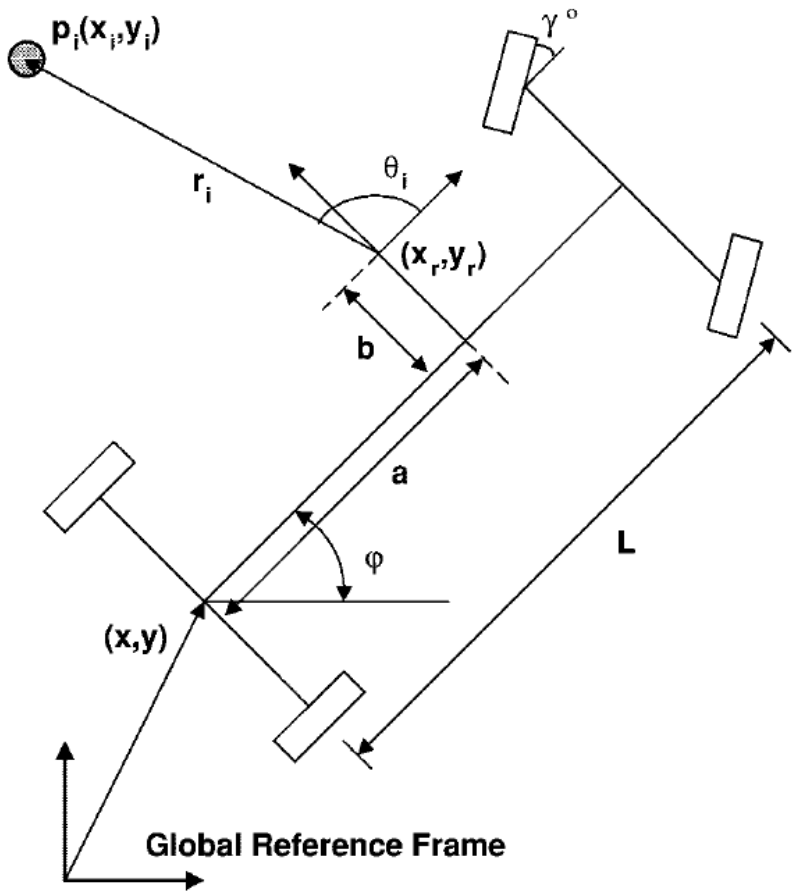
Wheel encoder examples

- Examples→Atlas→Wheel_distance

```
#include <ATLAS.h>

void setup() {
  initialise();
  zero_wheel();
}
unsigned long i;
void loop() {
  i = read_wheel();
  Serial.println(i);
}
```

Ackermann kinematics



$$\dot{x} = Vcos(\varphi)$$

$$\dot{y} = Vsin(\varphi)$$

$$\dot{\varphi} = \frac{Vtan(\gamma)}{L}$$

Keeping track of location

On each sample iteration:

$$\dot{x} = V cos(\varphi)$$

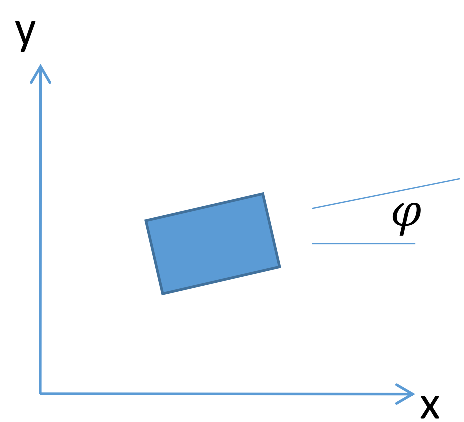
$$\dot{y} = V sin(\varphi)$$

$$\dot{\varphi} = \frac{V tan(\gamma)}{L}$$

Integrate to get the total position:

$$x = x_0 + \int_0^t \dot{x} dt$$

$$y = y_0 + \int_0^t \dot{y} dt$$



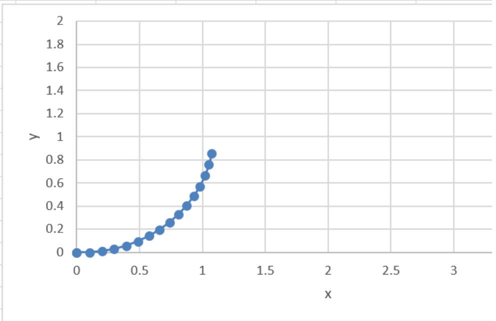
φ is determined by integrating rotational velocity:

$$\varphi = \varphi_0 + \int_0^t \dot{\varphi}\, dt$$

V – *velocity of vehicle*
φ – *speed of vehicle*
γ – *steer angle*
L – *vehicle length*

Forward kinematics Ack.xls

Velocity m/s	1	Steer angle		Length	0.6	Sample time secs	0.1
	$\dot{x} = V\cos(\theta)$	$\dot{y} = V\sin(\theta)$					
T	xdot	ydot		x	y	theta	
0	0	0		0	0	0	
0.1	1	0		0	0	0.096225045	
0.2	0.995373942	0.096076618		0.1	0	0.19245009	
0.3	0.981538567	0.191264324		0.199537	0.009608	0.288675135	
0.4	0.958621883	0.28468243		0.297691	0.028734	0.384900179	
0.5	0.926835917	0.375466621		0.393553	0.057202	0.481125224	
0.6	0.886474756	0.462776951		0.486237	0.094749	0.577350269	
0.7	0.837911828	0.545805615		0.574885	0.141027	0.673575314	
0.8	0.781596441	0.623784421		0.658676	0.195607	0.769800359	
0.9	0.718049632	0.6959919		0.736835	0.257986	0.866025404	
1	0.647859345	0.761759981		0.80864	0.327585	0.962250449	
1.1	0.571674987	0.82048017		0.873426	0.403761	1.058475494	
1.2	0.490201425	0.87160918		0.930594	0.485809	1.154700538	
1.3	0.404192462	0.91467396		0.979614	0.57297	1.250925583	
1.4	0.314443863	0.94927607		1.020033	0.664437	1.347150628	
1.5	0.221785993	0.975095366		1.051478	0.759365	1.443375673	
1.6	0.127076133	0.991892966		1.073656	0.856874	1.539600718	

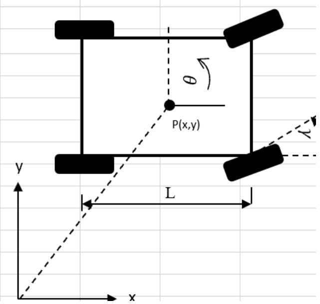


User Input:

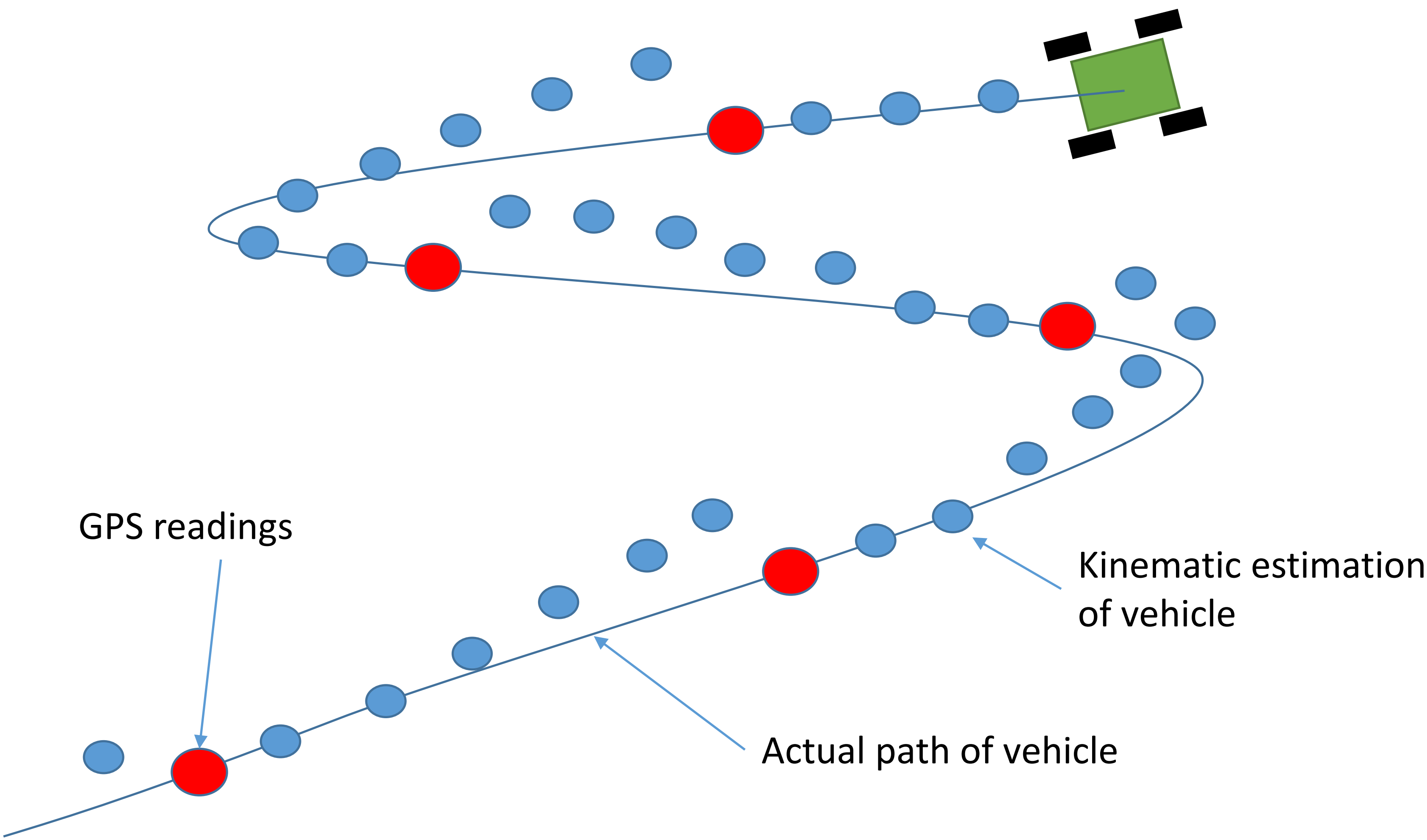
Velocity (m/s):			1
Steer angle degrees:			30
Vehicle Length			0.6

Calculated:

Steer angle radians:		0.523598776
Theta dot	$\dot{\theta} = \dfrac{V\tan(\gamma)}{L}$	0.962250449



What's the use of estimating x,y ?



Harper Adams University

Drive commands

drive (speed)   0-1 m/s, negative is backwards (-1)

This is open loop – soon we will close the loop using PID control

Tasks for this week

- Calibrate the steering so it centres well
- Have the robot drive in circles, measure the radius and calculate the actual steering angle
  - Compare for a range of angles
- Test the steer radius command and assess its accuracy
- Investigate the CAN steer messages on the TAFE tractor
- Use your kinematic knowledge to have the robot estimate its x,y location (use the Excel simulation to gain familiarity)