

# CO2削減を目的とした新交通システムの研究

## ～未来型自動車制御による渋滞のない交差点システムの開発～

追手門学院大手前高等学校 南方博 伊賀妃里

### 背景・目的

世界的に化石燃料から再生可能エネルギーへの転換が求められている。また、自動車はガソリン車からEV車に移行するだろう。しかし、再生可能エネルギーはエネルギー量も少なく、不安定である。そして、渋滞による問題はエネルギーだけでなく、大きな経済損失にも繋がっており、早期の解決が必要な課題である。

- ・渋滞損失時間の移動時間の約4割(年間約50億人・時間)
- ・経済損失約11兆円/年(国民総生産:GNPより2%多い)

これからの時代...

化石燃料

ガソリン車

再生可能エネルギー

EV車

再生可能エネルギーを有効活用する交通システム

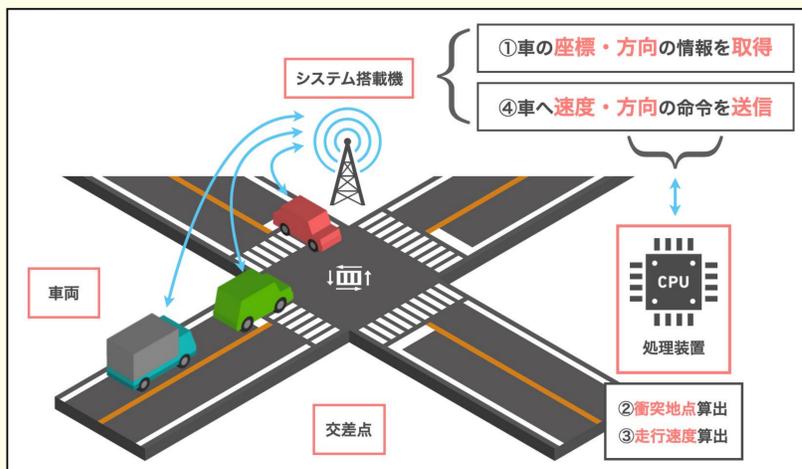
これらを解決するために...

エネルギーの損失の大きい渋滞を引き起こす交差点の渋滞に焦点をあて、「未来型自動車制御による新交通システム」の開発に挑戦した。

### 「No! 渋滞交差点システム」の構想

このシステムは、スクランブル交差点に行き交う多くの歩行者がお互いの距離を把握し、衝突せずに移動することをヒントに、オリジナルのアルゴリズムを考案した。

また、私たちが構想する交差点を右図に示す。これは、道路と車の通信を行うことで、実現できると考えた。



### 実装に向けたプログラム&数式

Bluetooth4.2を使用し、相互通信を行なった。また、アプリはUnityを使用。

```
1 //1台目のx座標  
Debug.LogFormat("pos=x:{0}", cube[i].pos.x);  
//1台目のy座標  
Debug.LogFormat("pos=y:{0}", cube[i].pos.y);  
//1台目の方向  
Debug.LogFormat("angle={0}", cube[i].angle);
```

① 交差点に居る全車両の位置・方向の情報を取得

C#言語でコーディング

```
2 //ロボから取得した方向の角度を弧度法に変換  
float tanA = Mathf.Tan(cube[0].angle * Mathf.PI / 180);  
float tanB = Mathf.Tan(cube[1].angle * Mathf.PI / 180);  
//ロボから取得した座標と角度から交点を計算  
float AnsX = (cube[0].pos.x * tanA - cube[1].pos.x * tanB) / (tanA - tanB);  
float AnsY = (-cube[1].pos.y * tanA + cube[0].pos.y * tanB) / (tanA - tanB);
```

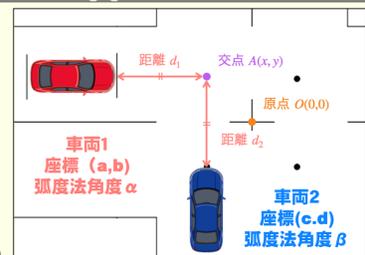
② 位置・方向の情報を元に衝突予想地点(交点A)を算出

```
3 //求められる交点を交差点内に限定  
if (AnsX > 0 && AnsX < 100 && AnsY > 0 && AnsY < 100)  
{  
    HighSpd = (int)(0.5 * d);  
}
```

③ 衝突予想地点を元に衝突しない走行速度を算出

```
4 //処理をリポート  
void Update()  
{  
    //衝突地点を求めると同時に衝突しない走行速度を算出  
    //衝突地点を求めると同時に衝突しない走行速度を算出  
    //衝突地点を求めると同時に衝突しない走行速度を算出  
}
```

④ 特定の車両の走行速度を算出された走行速度にする



交点Aから等距離の各車両と、アルゴリズムに使用する変数一覧

$$x = \frac{a \sin \alpha - c \tan \beta - b + d}{\tan \alpha - \tan \beta}$$

$$y = \frac{-c \tan \alpha \tan \beta + a \tan \alpha \tan \beta + d \tan \alpha - b \tan \beta}{\tan \alpha - \tan \beta}$$

・速度は一次関数的に減速する仕組みに

$$d_1 = \sqrt{(a-x)^2 + (b-y)^2}$$
$$d_2 = \sqrt{(c-x)^2 + (d-y)^2}$$
$$d_3 = |d_1 - d_2| \quad (d_3は100以下とする)$$
$$HighSpd = 0.5d_3$$

・衝突地点を求めるのに三角関数を使用

### 未来型自動車制御システムの概要

① すべての車の座標、車の方向・速度の情報を取得

② 座標・方向・速度から衝突予想地点を算出

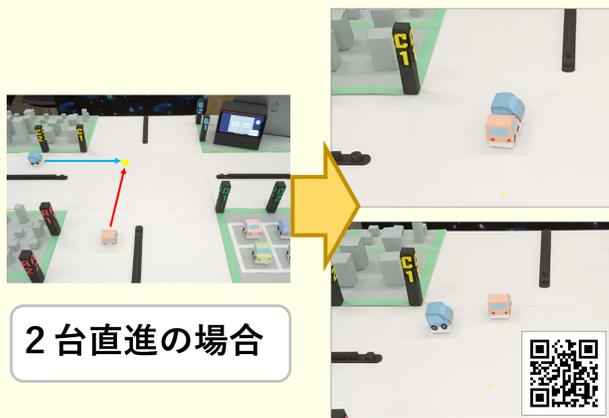
③ 衝突予想地点をもとに衝突しない走行速度を算出

④ それぞれの車に衝突を回避する走行速度を送信

左図の流れで、このシステムを実現させている

### (1) 2台の車で実装実験

開発したプログラムを用いて、実装実験を行った。交差点に2台の自動車を衝突点から同距離に配置。未来型自動車制御のあり・なしで比較実験を行った。結果は、下図のようになった。



### (2) 複数の車で実装実験

5台の実装実験を次に示す。(左折2台・直進3台・衝突地点3つの場合)

【プログラムの特徴(工夫点)】

- ・衝突が予想される場合、左側にいる車が減速する仕組みに。交差点内で衝突地点がある場合に限り減速させる。
- ・後続車が居る場合、前の車両の速度に合わせて減速、加速する。
- ・回避するまでの計算時間を早める為、メソッド(クラスや構造体)を活用し、少しでもラグを減らし、1つの処理につき0.02秒の高速化に成功した。



### 展望・考察

このシステムに実現によって、EV車の動力である再生可能エネルギーを無駄なく活用できるようになるだろう。さらに、渋滞をなくすことは、経済的な損失をなくし、私たちが失っていた時間を取り戻すことができ、より豊かな生活に繋がると考える。未来型自動車制御による新交通システムは、再生可能エネルギーを効率よく活用し、生活・交通の常識を大きく変えるエネルギー&交通革命を社会に提供できると期待する。