# G53ARS Formulae - Craig Knott

## Dc Motors

| | |
|---|---|
| Power | $P = \omega t$ |
| Speed | $\omega = K_V * V$ |
| Torque | $t = K_I * I$ |

## Sonar Sensors

Distance $D = \frac{ToF*Speed}{2}$

## Gearing

| | |
|---|---|
| Force | $F = t/r$ |
| Linear Speed | $v = \omega r$ |
| Output Speed | $\omega_2 = \frac{r1}{r2} * \omega_1$ |
| Output Torque | $t_2 = \frac{r2}{r1} * t_1$ |

## H-Bridge

| $T_l$ | $T_r$ | $B_l$ | $B_r$ | Effect |
|---|---|---|---|---|
| 1 | 0 | 0 | 1 | CW |
| 0 | 1 | 1 | 0 | CCW |
| 1 | 1 | 0 | 0 | Brake |
| 0 | 0 | 1 | 1 | Brake |

## PID Control

| Var | RT | OS | ST | SSE | STAB |
|---|---|---|---|---|---|
| P | - | + | S | - | - |
| I | - | + | + | E | - |
| D | S | - | - | N | + |

Continuous/Discrete Proportional, $K_p et$

Continuous Integral, $K_i \int_0^t e\tau d\tau$

Discrete Integral, $K_i \sum_{i=1}^{k} e(t_i)dt$

Continuous Derivative, $K_d \frac{det}{dt}$

Discrete Derivative, $K_d \frac{e(t_k)-e(t_{k-1})}{\Delta t}$

## Finite State Machine

DFA $D = \langle Q, \Sigma, \delta, q_0, F \rangle$

## Traversing a 2d Plane

Position, $\begin{bmatrix} x \\ y \\ z \end{bmatrix}$

New Position, $p' = p + \begin{bmatrix} \Delta S \; cos(\theta + \frac{\Delta\theta}{2}) \\ \Delta S \; sin(\theta + \frac{\Delta\theta}{2}) \\ \Delta\theta \end{bmatrix}$

Variation in Angle, $\Delta\theta = \frac{\Delta S_r - \Delta S_l}{B}$

Variation in Distance, $\Delta S = \frac{\Delta S_r + \Delta S_l}{2}$

## Kalman Filter

| | |
|---|---|
| Process Model | $x_k = A_k x_{k-1} + Bu_k + W_k$ |
| Measurement Model | $z_k = H_k x_k + v_k$ |
| Current State | $\hat{x}_{\bar{k}} = A\hat{x}_{k-1} + Bu_k$ |
| Error Covariance | $P_{\bar{k}} = AP_{\bar{k}}A^T + Q$ |
| Kalman Gain | $K_k = P_{\bar{k}}H^T(HP_{\bar{k}}H^T + R)^{-1}$ |
| E.C with KGain | $P_k = (I - K_k H)P_{\bar{k}}$ |
| Msrmnt with KGain | $\hat{x}_k = \hat{x}_{\bar{k}} + K_k(z_k - H\hat{x}_{\bar{k}})$ |
| Residual | $z_k - H\hat{x}_{\bar{k}}$ |

## Particle Filter

| | |
|---|---|
| Process Model | $x_k = f(x_{k-1}, u_k, w_k)$ |
| Measurement Model | $z_k = h(x_k, v_k)$ |
| A particle | $s_k^i = [x_k^i, w_k^i]$ |
| Update Particles | $x_k^i = f(x_{k-1}^i, u_k, w_k^i)$ |
| Update Weights | $w_k^i = w_k^i P(z_k \mid x_k^i)$ |
| Residual | $z_k - h(x_k^i, 0)$ |
| Weighted Mean | $\hat{x}_k = \sum_{i=1}^{N} x_k^i w_k^i$ |
| Best Particle | $\hat{x}_k = x_k^i \mid w_k^i = \max w_k^i$ |
| Robust Mean | $\hat{x}_k = \sum_{i=1}^{N} x_k^i w_k^i : \mid x_k^i - x_k^{best} \mid \leq \varepsilon$ |
| Eff. Sample Size | $N_{eff} = \frac{1}{\sum_{i=1}^{N} (w_k^i)^2}$ |
| Distance to point | $\sqrt{(x_a - x_b)^2 + (y_a - y_b)^2}$ |

Particle Updating Pseudo Code

```
for each time segment
  for each particle
    update state, using process model

  for each particle
    update weight, using msrmnt model

  normalise weights
  predict position
  if not suitably diverse
    resample
```

PID Control Code

```
last_error = setPoint - measurement
integral = 0
while ( true )
  wait(dt)
  error = setPoint - measurement
  integral = integral + error * dt
  derivative = (error - last_error) / dt
  output = (Kp * error)
           + (Ki * integral)
           + (Kd * derivative)
  last_error = error;
```