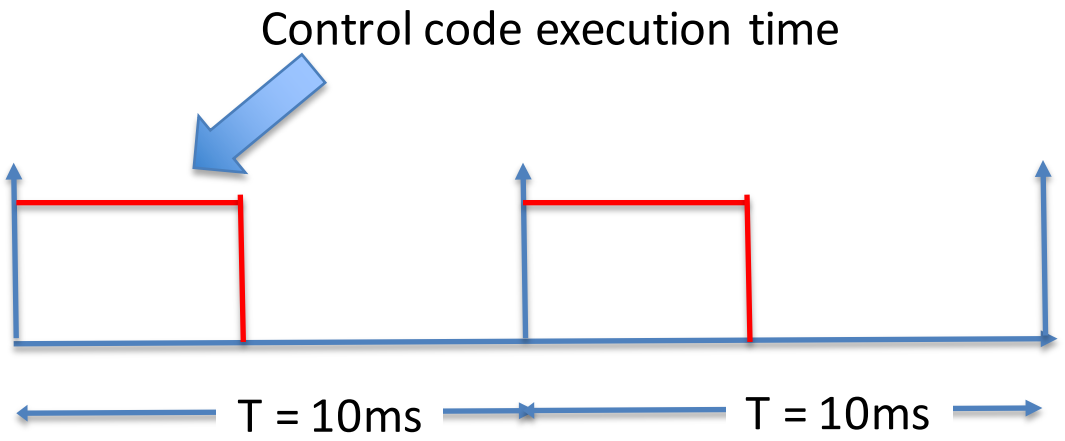
Real-time Interrupt (Timer)

Timer

- **Timer (Realtime Interrupt)**
  - Creates a deterministic execution of a function
  - For example, use to numerically determine velocity from position information (encoder)



Timer

- Deterministic function execution
  - Allows to accurately determine time derivative of a signal for example velocity

$$velocity\ [n] = (position[n] - position\ [n-1])/T$$

Current velocity = (current position – previous position)/T

Timer Implementation (Timer1)

```
void setup_timer(void);
```

The function setup_timer() creates an interrupt every 10ms and execute ISR(TIMER1_COMPA_vect)

```
/*The function below is coupled to timer1
interrupt*/


ISR(TIMER1_COMPA_vect)
// timer compare interrupt service routine
{
 control();
}
```

Additional Functions for Mobile Robot Control (mrobot.h)

Additional Functions that are executed every 10ms

```
void get_current_status(void);
```

The above function determines the current velocity of the wheels.

```
void low_level_control(void)
```

Computes the desired voltage and duty to track the desired velocity.

$$voltage = K_p(desired\ position - current\ position) + K_d(desired\ velocity - current\ velocity)$$

Low Level Control

- Based on PD (Proportional + Derivative) Control

$$voltage = K_p(desired\ position - current\ position) + K_d(desired\ velocity - current\ velocity)$$

- Given Vcc and for a 8 bit PWM, what is the expression for duty?

duty =

Sample Code

```
#include "mrobot.h”

void setup() {
setup_timer();

    /*This will setup an Interrupt Service Routine
      to be executed at 10ms*/

encoder_init();
motor_init();
Serial.begin(9600);
 }
void loop() {

}
```

Sample Code

```
void control(void){
get_current_status();
   /*****************************/
des_wvel[0] = 0.25;  // set motor 0 to 0.25 rad/sec
des_wvel[1] = 0.25;  // set motor 1 to 0.25 rad/sec
   /*****************************/
low_level_control();
}
/**This function will be executed every 10ms **/
ISR(TIMER1_COMPA_vect) /* timer compare interrupt
service routine*/
{
  control();
}
```