

**Feedrate Calculations for  
implementation with the LEGO NXT  
Servo Motors placed on  
rack-and-pinion system in LeJOS  
NXJ**

Papachristou Marios

January 2013

# Contents

<b>1</b>	<b>Feedrate Calculation Formula</b>	<b>2</b>
1.1	Implementation with the NXT Servo in LeJOS NXJ API . . .	2
<b>2</b>	<b>Defining the functions</b>	<b>4</b>

# Chapter 1

## Feedrate Calculation Formula

This formula can be used to figure out the feed rate that the cutter travels into or around the work. This would apply to cutters on a milling machine, drill press and a number of other machine tools. This is not to be used on the lathe for turning operations, as the feed rate on a lathe is given as feed per revolution:

$$F = f \cdot TC$$

where [1]:

1.  $F$  is feedrate calculated in *mm/min* or *in/min*
2.  $f$  is the calculated speed of the cutter (or the rotational frequency) measured in *RPM*
3.  $T$  is the number of teeth on the cutter (rack gear)
4.  $C$  is the chip load or feed per tooth. This is the size of chip that each tooth of the cutter takes.

### 1.1 Implementation with the NXT Servo in LeJOS NXJ API

LeJOS NXJ has a method to define speed: `NXTRegulatedMotor.setSpeed(int speed)` in *degrees/sec* which means that the feedrate measurement must be

adjusted in order to make the Servo Motor function properly [2]. So, the rotational frequency is calculated as **(1)**

$$f = \frac{F}{TC}$$

whereas frequency equals to **(2)**

$$f = \frac{\omega}{2\pi}$$

and it is measured in *rad/sec*. By converting rotational velocity to *degrees/sec* **(3)**

$$\omega = 1 \frac{rad}{sec} = 1 \frac{\pi}{180^\circ} \frac{degrees}{sec}$$

From **(1)**, **(2)** and **(3)** we end up to:

$$f = \frac{F}{TC} \tag{1.1}$$

$$\frac{\omega}{2\pi} = \frac{F}{TC} \tag{1.2}$$

$$\omega = \frac{2\pi F}{TC} \frac{rad}{sec} \quad or \quad \omega = \frac{\pi^2 F}{TC \cdot 90^\circ} \frac{degrees}{sec} \tag{1.3}$$

Moreover, we can treat  $\frac{2\pi}{TC}$  and  $\frac{\pi^2}{TC \cdot 90^\circ}$  as constants ( $c_1, c_2$ ).

$$\omega = c_1 \cdot F \quad or \quad \omega = c_2 \cdot F$$

## Chapter 2

### Defining the functions

Defining the methods was undoubtedly easier than the mathematical part

```
final double TEETH, CHIP_LOAD; //user defined constants

public static double getAngularVelocity (double feedrate) {
    final double c = (Math.PI * Math.PI) /
        (TEETH * CHIP_LOAD * 90);
    return (feedrate * c); // deg per sec
}

public static double getFeedrate (double angular_velocity) {
    final double c = (Math.PI * Math.PI) /
        (TEETH * CHIP_LOAD * 90);
    return (angular_velocity / c); // mm per min
}
```

NOTE: You must cast your results into integers by: `(int)(result)` so as they can be processable by the NXT brick.

# Bibliography

- [1] Smid, Peter (2003). "Feed Rate Equation". CNC Programming Handbook. Industiral Press, Inc.
- [2] LeJOS NXJ Application Programming Interface Documentation. Available on the web (28/5/2013) (<http://lejos.sourceforge.net/nxt/nxj/api/>)