

Design of Accurate Steering Gear Mechanism

V. CHITTI BABU
P. GOVINDA RAO
K. SANTA RAO

*Department of Mechanical Engineering, GMRIT, Rajam, Andhra Pradesh, India
govind09.p@gmail.com*

B. MURALI KRISHNA

Sri Sivani College of Engineering, Chilakapalem, Andhra Pradesh, India

Received (10 August 2017)

Revised (14 August 2017)

Accepted (13 September 2017)

This article focuses on the synthesis of a steering mechanism that exactly meets the requirements of steering geometry. It starts from reviewing the four-bar linkage, then discusses the number of points that a common four-bar linkage could precisely trace at most. After pointing out the limits of a four-bar steering mechanism, this article investigates the turning geometry for steering wheels and proposes a steering mechanism using servo motors and ARDUINO board. The pitch curves, addendum curves, dedendum curves, tooth profiles and transition curves of the noncircular gears are formulated and designed. Finally, kinematic simulations are executed to demonstrate the target of design

Keywords: steering, Ackermann, Davis, servo motor, Arduino.

1. Introduction

The steering gear mechanism is used for changing the direction of two or more of the wheel axles with reference to the chassis, so as to move the automobile in any desired path. Usually the two back wheels have a common axis, which is fixed in direction with reference to the chassis and the steering is done by means of the front wheels. In automobiles, the front wheels are placed over the front axles, which are pivoted at two points. These points are fixed to the chassis. The back wheels are placed over the back axle, at the two ends of the differential tube. When the vehicle takes a turn, the front wheels along with the respective axles turn about the respective pivoted points. The back wheels remain straight and do not turn. Therefore, the steering is done by means of front wheels only.

Saurabh Borse et al. [1] found that steering system is used for controlling the

directional characteristics and the stability of the vehicle. So in order to design and manufacture an efficient steering system one has to consider kinematic behavior of the system during its operation. Steering system of an ATV needs to be efficient as far as the parameters like weight, space considerations, rigidity are concerned. An ATV (All-Terrain Vehicle) is a vehicle which travels on all terrains. Because of this the steering system of an ATV is designed according to the specifications for the worst possible terrain or geographical profile. All the forces and torques encounter during the run are considered in order to design the mechanisms which will sustain these worst case scenarios. The concept of Yoke-nut assembly of the RPS that keeps the rack in proper mesh with the pinion is the main attraction of our project . Since the steering motions include frequent direction reversals as per the driving needs, any backlash is undesirable. Besides, the rack teeth portion is subjected to uneven wear. Hence, a Yoke-nut is fixed to the vehicle frame while the plunger rests against the rack with a spring between the Yoke-nut and plunger. A simulation and numerical analysis was performed and the result shows us the performance enhancement of the system because of using Yokenut assembly. Jing-Shan Zhao et al. [2] This article focuses on the synthesis of a steering mechanism that exactly meets the requirements of Ackermann steering geometry. It starts from reviewing of the four-bar linkage, then discusses the number of points that a common four-bar linkage could precisely trace at most. After pointing out the limits of a four-bar steering mechanism, this article investigates the turning geometry for steering wheels and proposes a steering mechanism with incomplete noncircular gears for vehicle by transforming the Ackermann criteria into the mechanism synthesis. The pitch curves, addendum curves, dedendum curves, tooth profiles and transition curves of the noncircular gears are formulated and designed. Kinematic simulations are executed to demonstrate the target of design.

R. S. Jadoun et al. [3] facilitate easy and proper steering of the vehicle while negotiating a turn, as per the requirements of driver in different situations like hair pin bends in hilly regions, sudden turns in highways and city streets or in sports car during race events an improved steering system is needed to address these problems. The steering arrangement used in a normal automobile was investigated and to solve the above problems a variable steering ratio mechanism was designed and fabricated, with the aim to determine if incorporation of variable steering ratio mechanism will make an improvement in the steady and transient state handling of the automobile. The size of the pinion gear and the number of teeth on the gear determine the rack-and-pinion steering ratio. The steering wheel must be turned one revolution to turn the front wheels one sixteenth of a turn, the steering ratio is 1 to 1/16. Reversing the numbers gives a ratio of 16 to 1, or 16:1. This steering ratio is always fixed. A variable steering ratio mechanism was fabricated by using constant mesh type gears from Bajaj-Super and steering mechanism from Tata-Nano. Gear housing using wood is created and gear shifting arrangement incorporated. It was observed that on engaging the 1st gear steering ratio is increased to 9.92:1. On engaging the 2nd gear steering ratio is increased to 7.20:1, On engaging the 3rd gear steering ratio is increased to 4.96:1 On engaging the 4th gear steering ratio is increased to 3.52:1. The effort required for steering is increased due to these ratios as energy is lost due to friction. This mechanism can only be practical if used in conjugation with power assist.

N. Laxmi et al. [4] special safety features have been built into cars for years, some for the safety of car's occupants only, and some for the safety of others. One of the choices available is Design and fabrication of steering controlled head light system. Car safety is the avoidance of automobile accidents or the minimization of harmful effects of accidents, in particular as pertaining to human life and health. Still, more specially, this device relates to a headlight arrangement operably connected to the steering and front wheel assembly of and automobile operably to maintain headlight members and the front wheels pointed in the same direction at all times.

Boby George et al. [5] made an attempt for designing, analysing, fabrication and testing of steering and braking system for a student Baja car and their integration in the whole vehicle. The car has been designed and fabricated to the best of its possible. The primary objective of this project was to identify and determine the design the parameters of a vehicle with a proper study of vehicle dynamics. This project helped us to study and analyse the procedure of vehicle steering and braking system designing and to identify the performance affecting parameters. It also helped us to understand and overcome the theoretical difficulties of vehicle design.

A. K. W. Ahmed et al. [6] found that a fixed Ackerman steering ratio between left and right wheels is not ideal for all speeds since slip angles are developed which is further influenced by the available normal load on the wheels. This is addressed by Active Front Steering (AFS) systems that actively alter the steering angle based on driver command, and corresponding response. Such corrections introduced to both wheels simultaneously without consideration of normal load cannot maximize the tire's ability to generate lateral forces. An Active Front Independent Steering (AIFS) system with capability for control of the wheel's steer angles independently can overcome this limitation. The concept of steer-by-wire although capable of providing such control, is not considered a fail-safe system. An innovative design of a mechanism for AIFS control is proposed in this paper. This design based on tandem planetary gear configuration is shown to be fail-safe system for AIFS along with many other potential applications.

Based on the past literature, most of the researchers concentrated to reduce the possibility of skidding, when a vehicle takes a turn. But, in the present work authors made an attempt to avoid skidding completely when vehicle takes a turn using Aurdino board and servomotors run by computer programming.

2. Principle

In order to avoid skidding (i.e. slipping of the wheels sideways), the two front wheels must turn about the same instantaneous centre I which lies on the axis of the back wheels as shown Fig. 1. If the instantaneous centre of the two front wheels do not coincide with the instantaneous centre of the back wheels, the skidding on the front or back wheels will definitely take place, which will cause more wear and tear of the tyres. Thus, the condition for correct steering is that all the four wheels must turn about the same instantaneous centre. The axis of the inner wheel makes a larger turning angle than the angle subtended by the axis of outer wheel.

Let a – wheel track, b – wheel base, and c – distance between the pivots A and B of the front axle, $\cot \Phi - \cot \theta = c/b$.

2. When the vehicle is steering to the right, the position of the gear is shown by firm lines. In this position, the lines of the front wheel axle intersect on the back wheel axle at I, for correct steering.
3. When the vehicle moves along a straight path the longer links AB, PQ are suitably proportioned. The value of θ and ϕ may be obtained either graphically or by calculations.

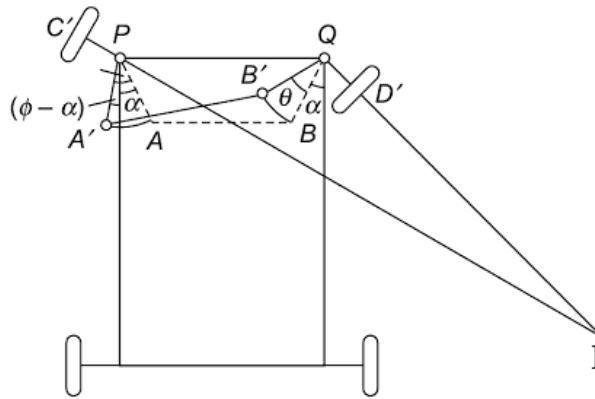


Figure 2 Ackermann Steering Gear Mechanism [7]

5. Components required for developing correct steering

5.1. Arduino

Arduino is an open source, computer hardware and software company, project, and user community that designs and manufactures microcontroller kits for building digital devices and interactive objects that can sense and control objects in the physical world. The project's products are distributed as open-source hardware and software, which are licensed under the GNU Lesser General Public License (LGPL) or the GNU General Public License (GPL), [1] permitting the manufacture of Arduino boards and software distribution by anyone. Arduino boards are available commercially in preassembled form, or as do-it-yourself kits.

Arduino board designs use a variety of microprocessors and controllers. The boards are equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The boards feature serial communications interfaces, including Universal Serial Bus (USB) on some models, which are also used for loading programs from personal computers. The microcontrollers are typically programmed using a dialect of features from the programming languages C and C++. In addition to using traditional compiler tool chains, the Arduino project provides an integrated development environment (IDE) based on the Processing language project.

In general an arduino board is like any other microcontroller board but unlike them, it comes with own boot loader (a piece of code that runs when the controller

is powered up) which makes it possible to program the arduino directly from the computer, making it compact and self-sufficient.

Arduino programs are written in C/C++. The Arduino IDE comes with lots of software/hardware library. The files make many common input/output operations much easier. And it's easy to program even for a beginner. The Arduino programming environment is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well.



Figure 3 Arduino

5.2. Servo motors

A servomotor shown in below Fig. 3 is a rotary actuator or linear actuator that allows for precise control of angular or linear position, velocity and acceleration. It consists of a suitable motor coupled to a sensor for position feedback. It also requires a relatively sophisticated controller, often a dedicated module designed specifically for use with servomotors. Servomotors are not a specific class of motor although the term servomotor is often used to refer to a motor suitable for use in a closed-loop control system. Servomotors are used in applications such as robotics, CNC machinery.

6. Methodology

As we can observe from the history, we came across the types of steering gear mechanisms out of where Davis Steering gear mechanism works with sliding pairs and Ackermann Steering Gear mechanism satisfies the correct condition for steering only at three points. As an advanced point of approach we managed to satisfy the correct condition for steering using some programming given to arduino. A Steering Gear Mechanism is first constructed with the help of servo motors. An arduino board is arranged along with the same construction to help the operator operate the whole system as shown below. The following is the program given to the arduino to obtain the correct condition for steering.



Figure 4 Servomotors

The Main Algorithm for condition for Correct Steering is:

```
int angle(int v){
x=(M_PI*v)/180;
d=tan(x);
y=atan((1*d)/(1+(w*d)));
z=(y*180)/M_PI;
return z;
}
```

This is derived from equation:

$$\text{Cot } \varphi - \text{Cot } \theta = w/l$$

PROGRAM FOR TRANSMITTER

```
#include<SPI.h>
#include<nRF24L01.h>
#include<RF24.h>
#include <printf.h>
#include "Wire.h"
#include "I2Cdev.h"
#include "MPU6050_6Axis_MotionApps20.h"
//#include "MPU6050.h" // not necessary if using MotionApps include file
MPU6050 mpu;
bool blinkState = false;
float angle_pitch_output,anglep,angler, angle_roll_output;
byte transpitch,transroll,strangle;
// MPU control/status vars
bool dmpReady = false; // set true if DMP init was successful
uint8_t mpuIntStatus; // holds actual interrupt status byte from MPU
uint8_t devStatus; // return status after each device operation
(0 = success, !0 = error)
uint16_t packetSize; // expected DMP packet size (default is 42 bytes)
uint16_t fifoCount; // count of all bytes currently in FIFO
uint8_t fifoBuffer[64]; // FIFO storage buffer
// Orientation/motion variables
```

```

Quaternion q;
VectorFloat gravity;
float euler[3];
float ypr[3];
// Use the following global variables and access functions to help store the overall
// rotation angle of the sensor
unsigned long last_read_time;
float last_x_angle; // These are the filtered angles
float last_y_angle;
float last_z_angle;
float last_gyro_x_angle; // Store the gyro angles to compare drift
float last_gyro_y_angle;
float last_gyro_z_angle;
void set_last_read_angle_data(unsigned long time, float x, float y, float z, float x_gyro,
float y_gyro, float z_gyro) {
last_read_time = time;
last_x_angle = x;
last_y_angle = y;
last_z_angle = z;
last_gyro_x_angle = x_gyro;
last_gyro_y_angle = y_gyro;
last_gyro_z_angle = z_gyro;
}
inline unsigned long get_last_time() {return last_read_time;}
inline float get_last_x_angle() {return last_x_angle;}
inline float get_last_y_angle() {return last_y_angle;}
inline float get_last_z_angle() {return last_z_angle;}
inline float get_last_gyro_x_angle() {return last_gyro_x_angle;}
inline float get_last_gyro_y_angle() {return last_gyro_y_angle;}
inline float get_last_gyro_z_angle() {return last_gyro_z_angle;}
// Use the following global variables
// to calibrate the gyroscope sensor and accelerometer readings
float base_x_gyro = 0;
float base_y_gyro = 0;
float base_z_gyro = 0;
float base_x_accel = 0;
float base_y_accel = 0;
float base_z_accel = 0;
int th = A0;
int thValue = 0;
// This global variable tells us how to scale gyroscope data
float GYRO_FACTOR;
// This global variable tells how to scale accelerometer data
float ACCEL_FACTOR;
// Variables to store the values from the sensor readings
int16_t ax, ay, az;
int16_t gx, gy, gz;

```



```

// Buffer for data output
char dataOut[256];
const uint64_t pipeOut = 0xE8E8F0F0E1LL;
RF24 radio(7,8);
struct MyData {
byte angle;
byte pitch;
byte f1;
byte f2;
byte f3;
byte f4;
};
MyData data;
void resetData()
{
data.angle = 0;
data.pitch = 127;
data.f1 = 0;
data.f2 = 0;
data.f3 = 0;
data.f4 = 0;
}
// =====
// ==== INTERRUPT DETECTION ROUTINE ====
// =====
volatile bool mpuInterrupt = false; // indicates whether MPU interrupt
pin has gone high
void dmpDataReady() {
mpuInterrupt = true;
}
// =====
// ==== CALIBRATION ROUTINE ====
// =====
// Simple calibration - just average first few readings to subtract
// from the later data
void calibrate_sensors() {
int num_readings = 10;
// Discard the first reading (don't know if this is needed or
// not, however, it won't hurt.)
mpu.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);
// Read and average the raw values
for (int i = 0; i < num_readings; i++) {
mpu.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);
base_x_gyro += gx;
base_y_gyro += gy;
base_z_gyro += gz;
base_x_accel += ax;

```

```

base_y_accel += ay;
base_y_accel += az;
}
base_x_gyro /= num_readings;
base_y_gyro /= num_readings;
base_z_gyro /= num_readings;
base_x_accel /= num_readings;
base_y_accel /= num_readings;
base_z_accel /= num_readings;
}
// =====
// ==== INITIAL SETUP ====
// =====
void setup() {
pinMode(4,INPUT_PULLUP);
pinMode(5,INPUT_PULLUP);
pinMode(6,INPUT_PULLUP);
pinMode(9,INPUT_PULLUP);
pinMode(A0,OUTPUT);
pinMode(A1,OUTPUT);
pinMode(A3,OUTPUT);
pinMode(A2,INPUT);
digitalWrite(A3,HIGH);
digitalWrite(A1,LOW);
digitalWrite(A0,LOW);
// join I2C bus (I2Cdev library doesn't do this automatically)
Wire.begin();
radio.begin();//Start I2C as master
radio.setChannel(??);
radio.setPALevel(RF24_PA_MAX);
radio.setAutoAck(false);
radio.setDataRate(RF24_250KBPS);
radio.openWritingPipe(pipeOut);
resetData();
// initialize serial communication
// Serial.begin(57600);
while (!Serial); // wait for Leonardo enumeration, others continue immediately
// NOTE: 8MHz or slower host processors, like the Teensy @ 3.3v or Arduinio
// Pro Mini running at 3.3v, cannot handle this baud rate reliably due to
// the baud timing being too misaligned with processor ticks. You must use
// 38400 or slower in these cases, or use some kind of external separate
// crystal solution for the UART timer.
// initialize device

```

7. Conclusions

The idea behind the present work is to design a steering gear mechanism which steers through any required angle so that the ideal condition for steering must be achieved. The developed steering mechanism basically helps to reduce the efforts and space required for a person to steer the vehicle. This is established using servo motors and arduino board which are easy to set up, and highly efficient in attaining outputs. Components used in this system are easy to manufacture, material used is feasible, reliable and easily available in market. An advanced program on arduino can be used thereby making the universal rule of steering satisfied.

References

- [1] **Borse, S., Chille, P., Dabhade, S., Deshmukh, S., Kulkarni, A.:** Design and Manufacturing of an effective steering system for ATV and performance enhancement using yoke-nut assembly, *special issue from national conference (NCMMM) in International Journal of Research in Advent Technology*, **2016**.
- [2] **Zhao, J.-S., Liu, X., Feng, Z.-J., Dai, J. S.:** Design of an Ackerman-type steering mechanism, *Proceedings of Institution of Mechanical Engineering Part C: Journal of Mechanical Engineering Science*, 227, 11, 2549–2562, **2013**.
- [3] **Jadoun, R. S., Kumar Choudhary, S.:** Design and fabrication of variable steering ratio mechanism for light Motor vehicle, *Innovative Systems Design and Engineering*, 5, 9, 7–15, **2014**.
- [4] **Laxmi, N., Anil Kaumar, B., Kiran Varma, D.:** Design and Fabrication of A Steering Controlled Headlights in Automobile, *International Journal & Magazine of Engineering, Technology, Management and Research*, 2, 7, 98–102, **2015**.
- [5] **Boby, G., Akhil, T. B., Albert, J., Aswin, J., Denny, F.:** Design and Fabrication of Steering and Bracking System for All Terrain Vehicle, *International Journal of Scientific & Engineering Research*, 7, 3, 7–18, **2016**.
- [6] **Ahmed, A. K. W., Rawat, V., Bhat, R. B.:** Vehicle Steering Mechanism for Active Independent Front Steering System, *3rd International Conference on Mechanical, Production and Automobile Engineering (ICMPAE'2013) January 4th to 5th Bali (Indonesia)*, 83–85, **2013**.
- [7] **Rattan, S. S.:** Theory of machines, *Tata McGraw-Hill Education*, **2014**.

