Intel Research

laboratory @ seattle

University of Washington Computer Science & Engineering

CSE 481

Robot Blimp Capstone

UNIVERSITY OF WASHINGTON

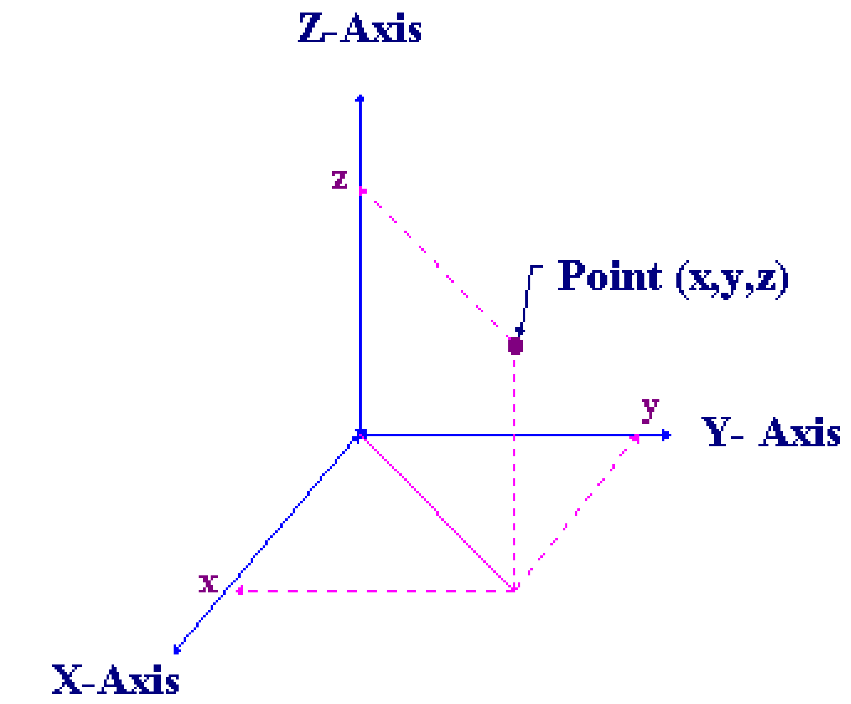
General Rules

Metric:

- All distances in meter (m)
- All times in seconds (s)
- All angles in radians (rad)

General Rules (2)

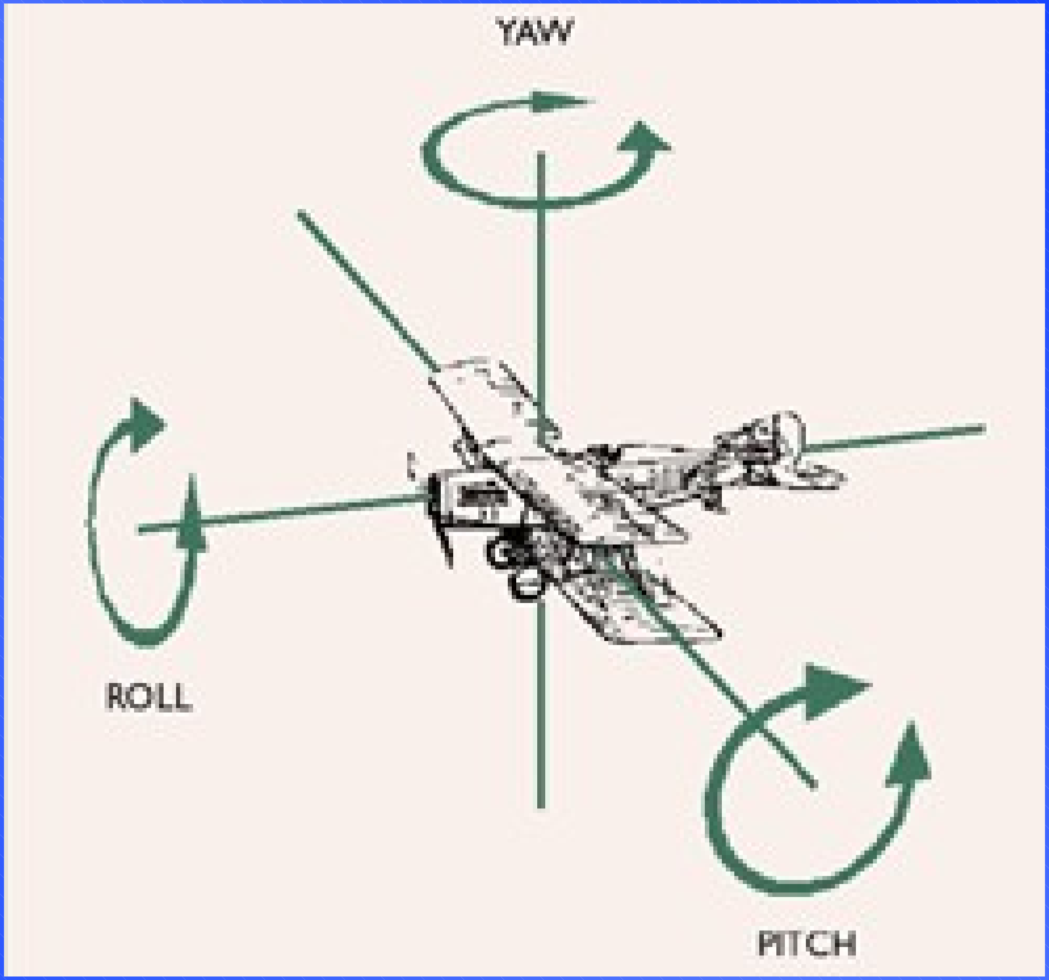
3D Coordinate System:



Three-Dimensional Cartesian Coordinates
X, Y, Z

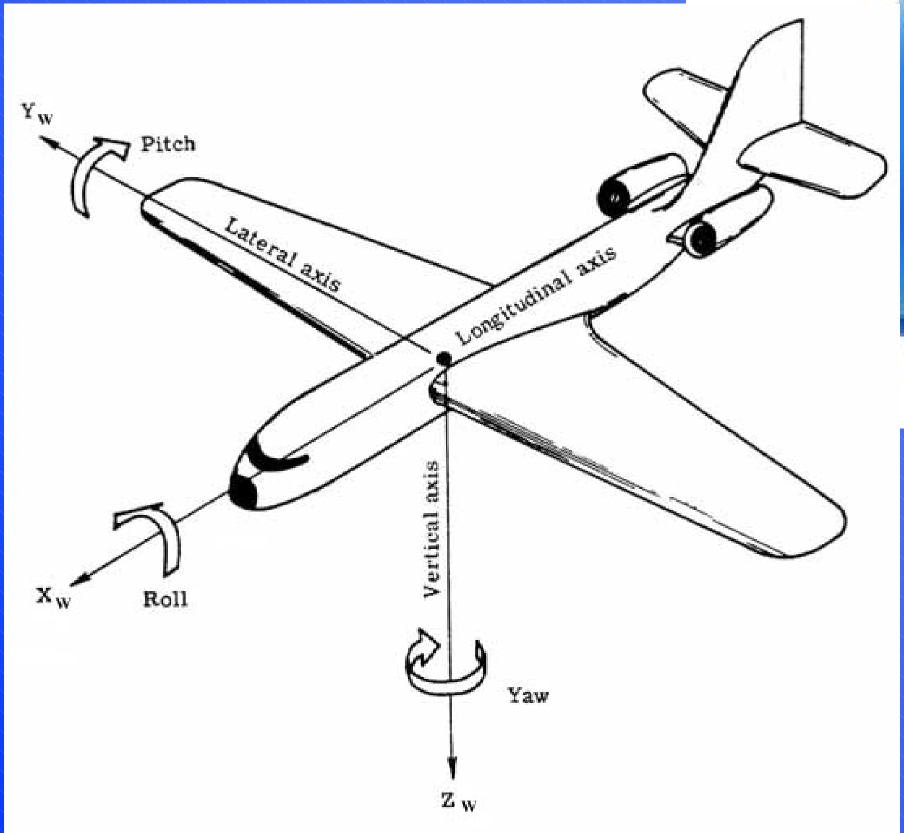
General Rules (3)

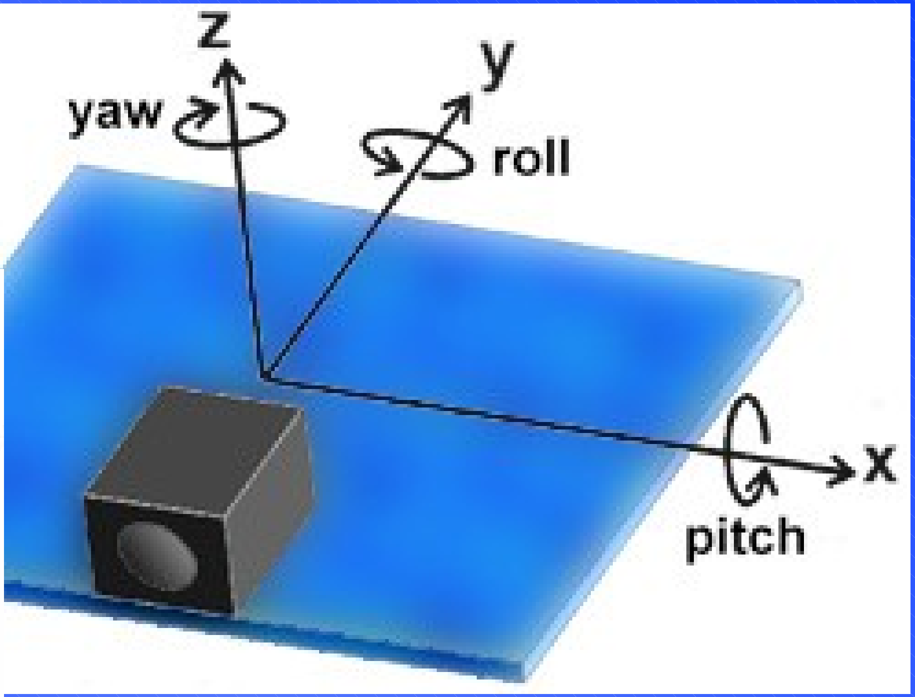
Yaw, Pitch, Roll



General Rules (4)

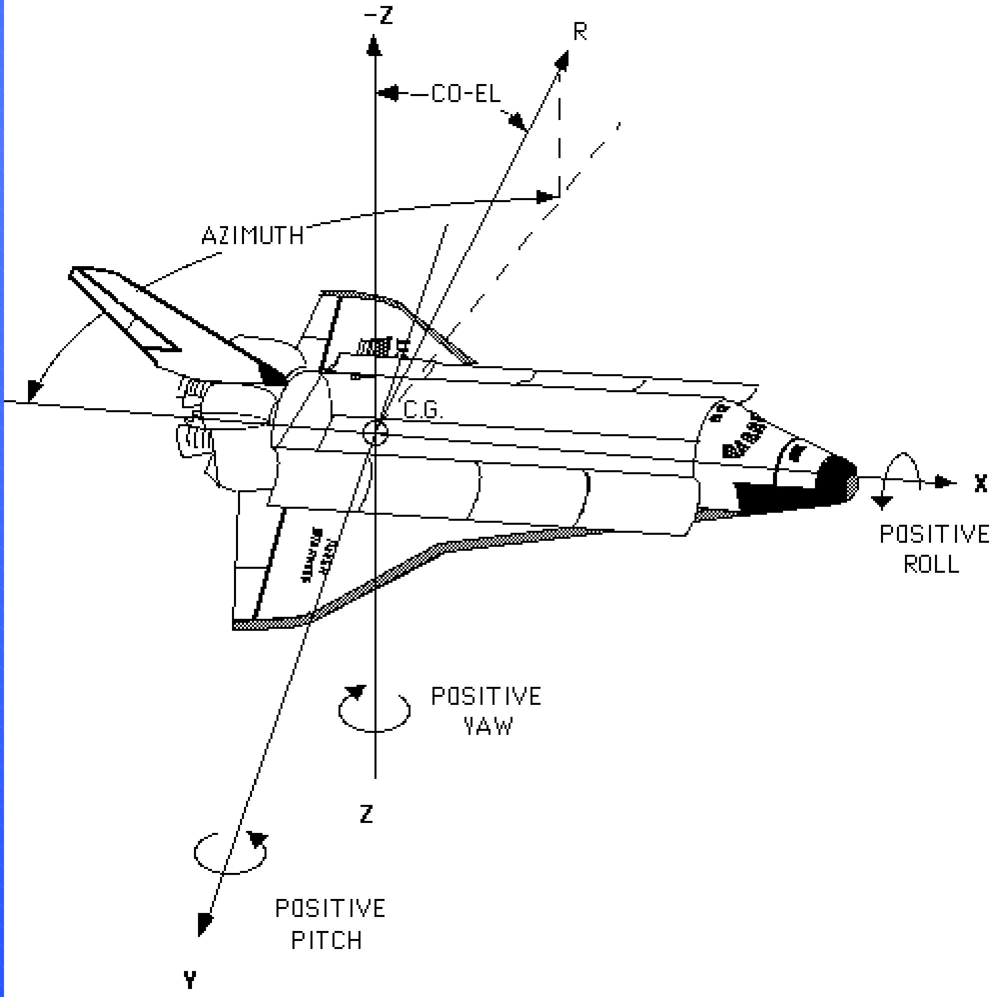
Be consistent !





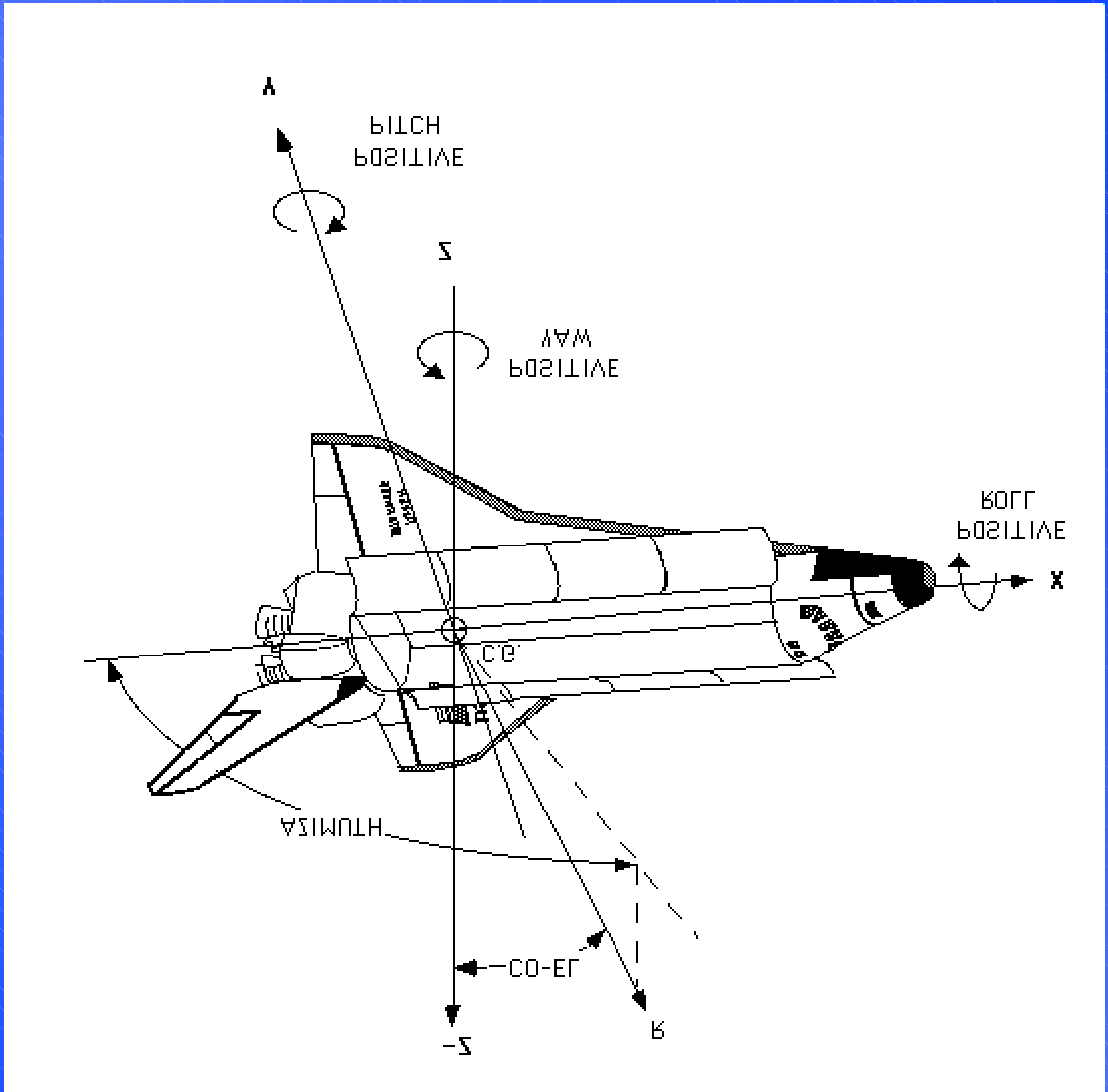
General Rules (5)

Most used:



General Rules (6)

We use:



General Rules (7)

Yaw = 0 ->

Yaw = π/2 ->

Yaw = - π/2 ->

Pitch = 0 ->

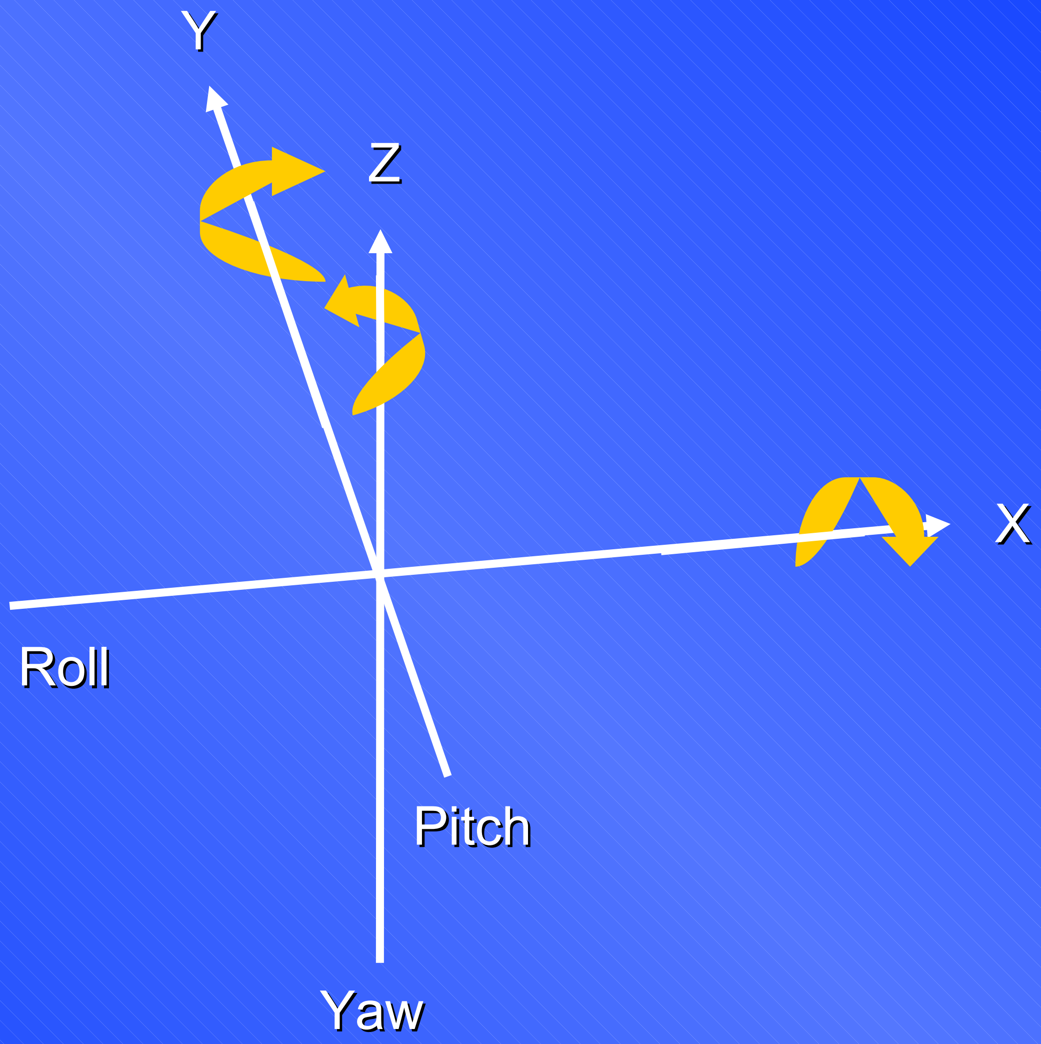
Pitch = π/2 ->

Pitch = - π/2 ->

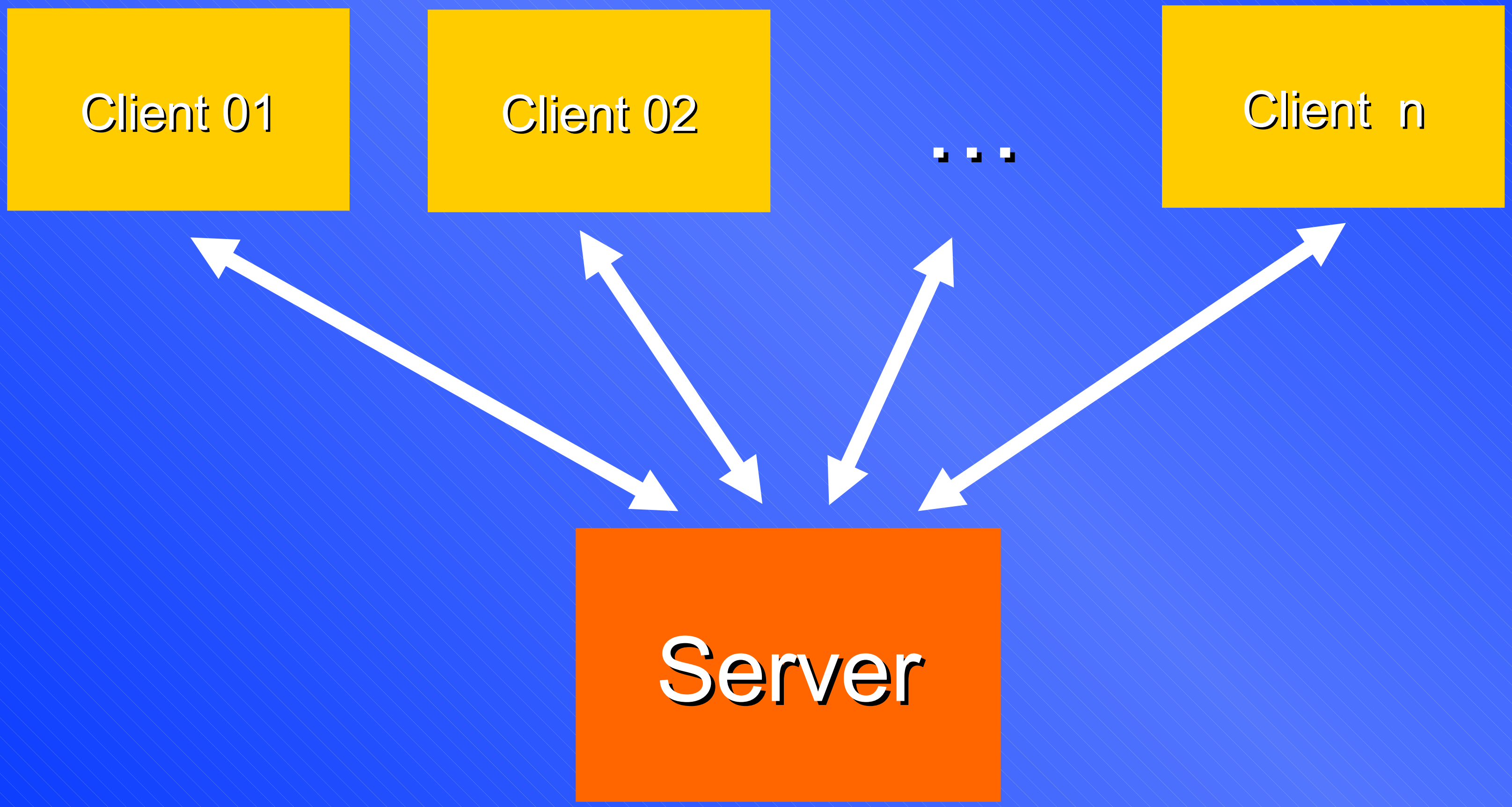
Roll = 0 ->

Roll = π/2 ->

Roll = - π/2 ->



IPC (www.cs.cmu.edu/~ipc)



Client 01

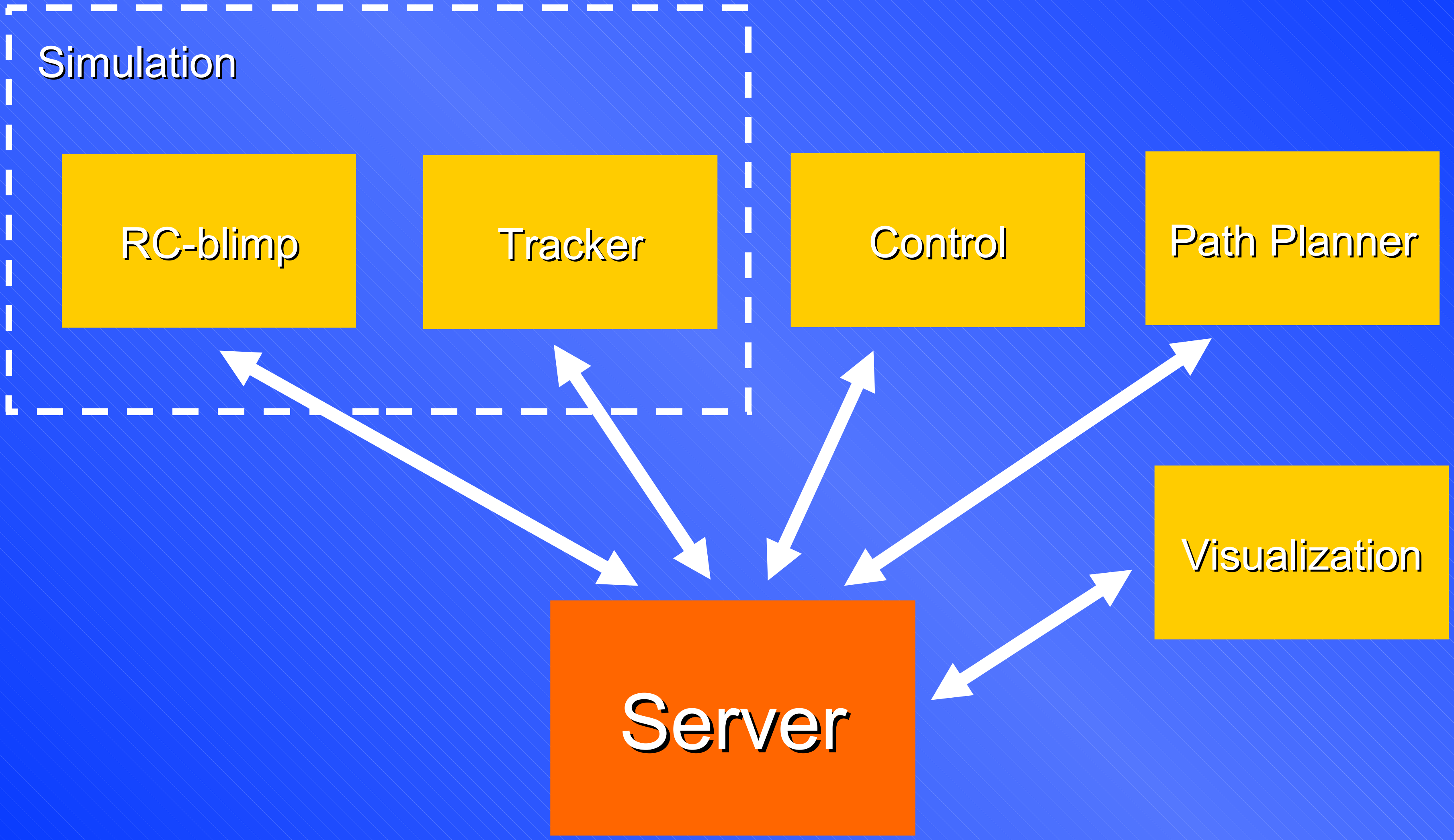
Client 02

. . .

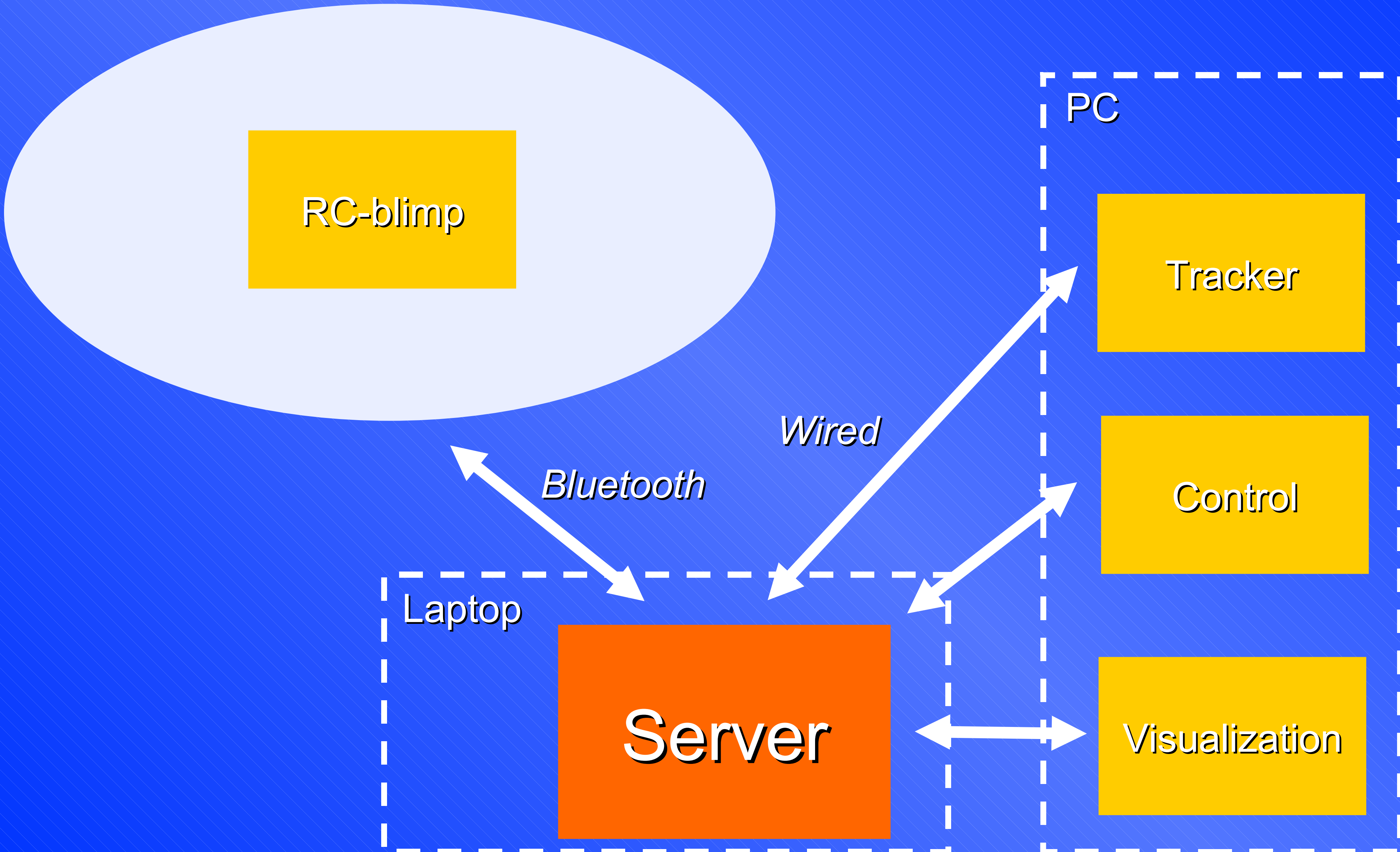
Client n

Server

Blimp Architecture



Blimp Architecture



IPC (www.cs.cmu.edu/~ipc)

- Messages are always defined in header files *name*_messages.h:

```
typedef struct {
        double timestamp;
        …
        …
} blimp_name_type_message;

#define BLIMP_name_type_FMT “{ double, … }”
#define BLIMP_name_type_NAME “blimp_name_type”
```

IPC (www.cs.cmu.edu/~ipc)

- Always timestamps !!!

```
blimp_name_type_message msg;
msg.timestamp = blimp_get_time();
```

In Java: msg.timestamp = System.currentTimeMillis()*0.001

- We can record all messages in a single file …
  - logger <blf-file>
- We can replay the same actions
  - playback <blf-file>

Logging (BLF: Blimp Log File)

- Text (ascii) file
- Single line for single message
- Always same structure:
  - KEYWORD TIMESTAMP data …
- Keywords:
  - BLIMP-STATUS : message from rc-blimp
  - BLIMP-CONTROL: message to rc-blimp
  - BLIMP-LOCALIZE: message from tracker