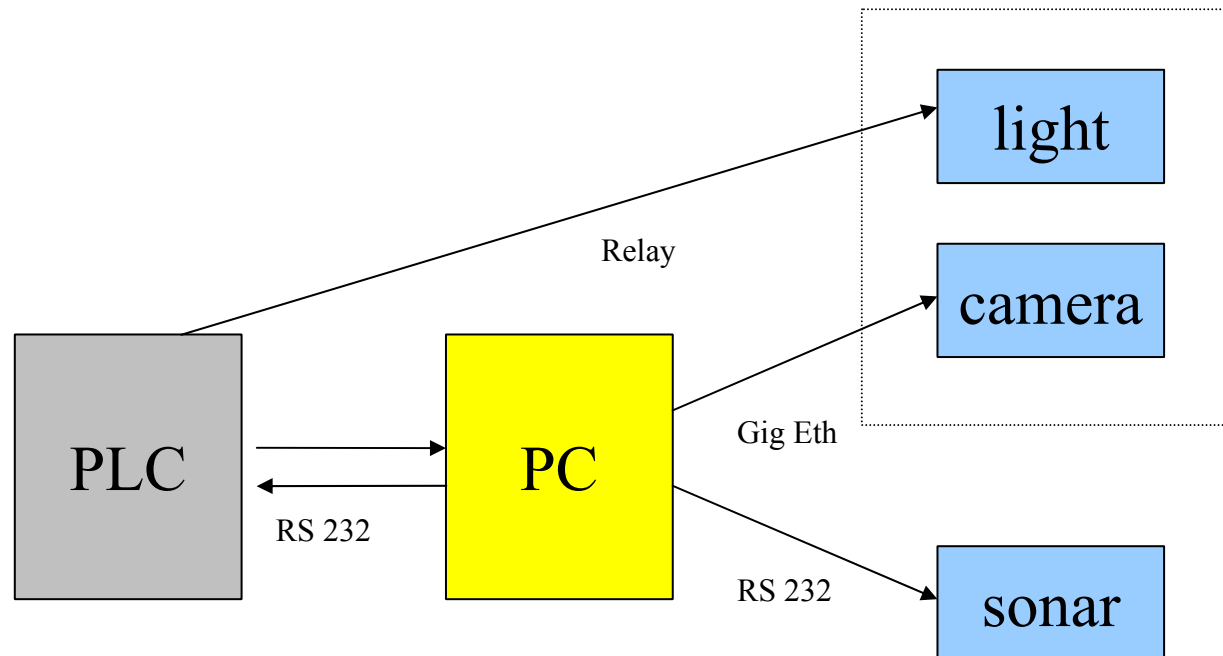# WP3
Automatic positioning for load /unload

Maciej Pakulski
Speed Poland

SYSTEM ARCHITECTURE

HARDWARE

Cameras:

- Pixelink, CMOS, 1280x1024, GigE monochromatic

Sonar

- 4-20mA analog output, converted in RS232
- 0-3m working distance (programmable)

HARDWARE

Illumination:
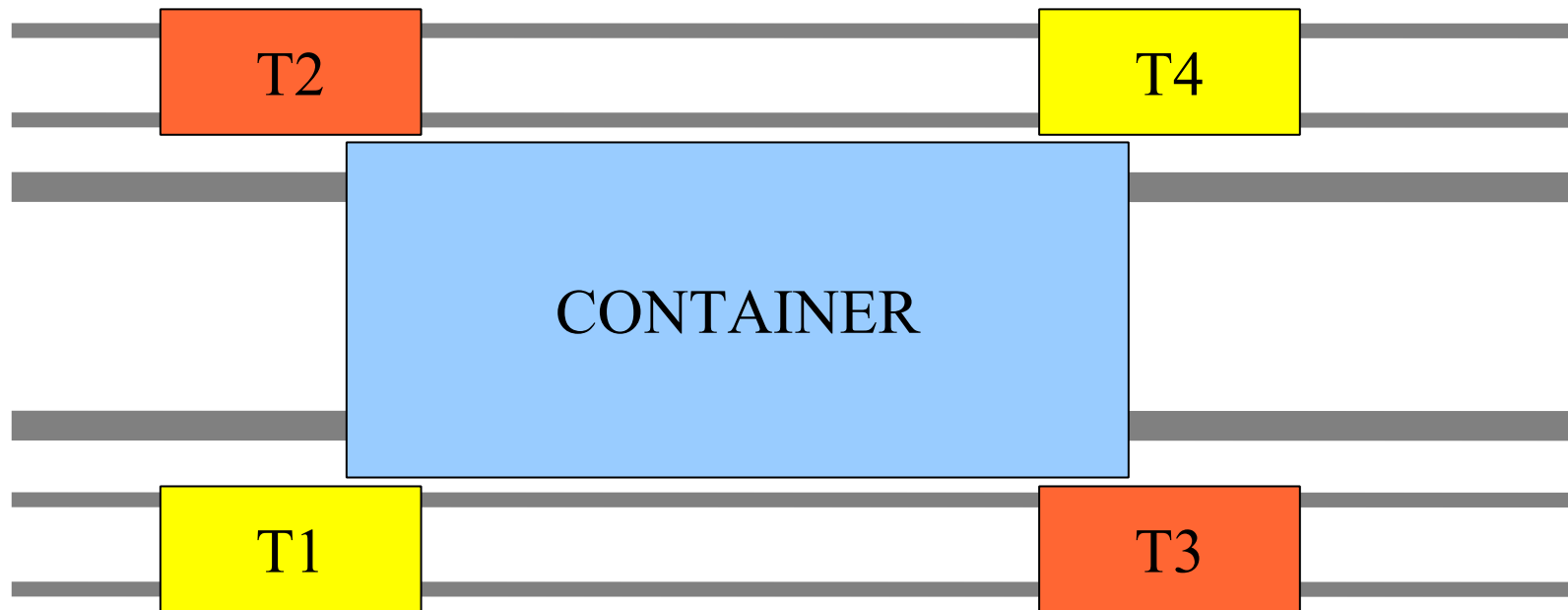- Halogen bulb, 500W

MACHINE PURPOSE

Automatic Unloading:
- from wagon to storage
- "hole finding procedure"= illumination + cam + pc

Automatic Loading:
- from storage to wagon
- "retaining pin finding procedure" = sonar + pc
- "hole finding procedure"= illumination + cam + pc

MACHINE LAYOUT

T2

T4

CONTAINER

T1

T3

PC ↔ PLC HANDSHAKE

PLC sends some commands through the RS232 interface
to the PC and receives the status or the coordinates:

- camera init

- start grab

- end grab

- sonar init

- start sonar

- stop sonar

Automatic Unloading Procedure

Automatic Unloading Cycle:
1. PLC moves the turret to a first position (x, y) - prepositioning
2. PLC → camera init to PC
       PC switch on the camera and tunes the shutter time
3. PLC → start grab to PC
       PC grabs and computes the coordinates (x0, y0) of the hole
4. PLC moves to (X=x+x0, X=y+y0)
5. PLC → start grab to PC
       PC grabs and computes the coordinates (x1, y1) of the hole
6. PLC moves to (X=X+x1, Y=Y+y1)

The cycle stops when the coordinates (xn, yn) from the camera are close to (0,0).

Automatic Unloading Procedure

After this, the PLC moves the finger (z direction) till the magnetic sensor mounted over the finger is on; if the motor-drive gets an alarm without having on the sensor, then there's an error from the camera – we have a collision.

If the camera is not able to compute any hole, it communicates a message "ERROR" to the PLC. If this happens, Metrocargo can extrapolate the position of the hole from the other camera on the same side of the container.

Automatic Loading Procedure

Automatic Loading Cycle:
1. PLC moves the turret to a first position (x, y) along the wagon
2. PLC → sonar init to PC
    PC connects to the sonar
3. PLC → start sonar to PC and start to move along the wagon
    PC reads all the z coordinate from the sonar
    As soon as it finds a local minimum, PC communicates
    that in that position there's a pin

In the case the position of the retaining pin is not equal to the container length, PLC can also use the camera to compute the position of the pin

Automatic Loading Procedure

Cameras are also used to compute the coordinates of the corner of the container also for loading procedure. This is to be sure to load the container in the correct position over the pins.

This cycle is similar to the unloading procedure phase.

UNLOADING RELABILITY

Some data about reliability

Number of unloading cycle = 430
Error from T1 = 2 (0.4 % - A = 99.6%)
Error from T2 = 5 (1.1 % - A = 98.9%)
Error from T3 = 7 (1.6 % - A = 98.4%)
Error from T4 = 0 (0.1 % - A = 99.9%)

System stops due to the cameras = 0

A= min (100-(E1*E4), 100-(E2*E3)) = min (99.96, 98.24) = 98.2%

LOADING RELIABILITY

Some data about reliability

Number of loading cycle = 430
Error from T1 = 0
Error from T2 = -
Error from T3 = -
Error from T4 = 1

System stops due to the vision system = 1

A = 99.8%

VI ALGORYTHM

Vision System algorythm:
- based on OpenCV library
- written in C++, developed using Microsoft Visual Studio

1. set exposure time high (0.1 s)
2. grab the picture and find a blob (the hole)
- thresholding (filter to avoid noise fromt the image)
- Canny edge detection to obtain the contours
- the contour is a hole only if lenght, compactness,
    width, height are within prefixed parameters
3. compute the hole center position in mm, using the pixel/mm
    factor (known by a camera calibration)

VI ALGORYTHM

4. communicate to PLC the correction between the hole center and the optical axis center in mm

5. if the hole is aligned to the camera optical axis, then the correction is (0, 0)

6. Turret moves using a fixed offset to correct the position of the camera to the position of the finger

7. If camera doesn't detect any hole, decrease the shutter time and try again (shutter time step = 0.05 s)
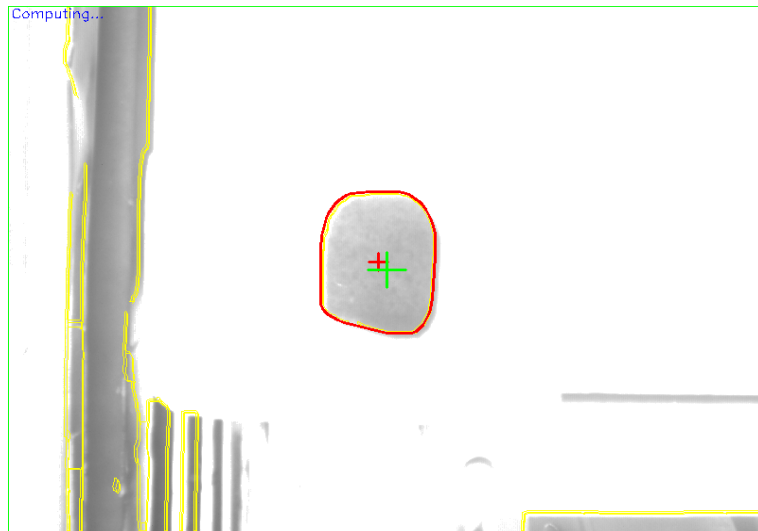
VI ALGORYTHM

VI ALGORYTHM

VIT PROJECT

Thank you