Control levels

In order to determine what approach will work best, it's needed to be aware of the following control hierarchy:

4	Logistic control level	Planning sequences to achieve a final goal: Load all the boxes from the stockpile on the left upon the palette on the right.
3	Sequence control level	Chaining atomary commands to achieve an intermediate goal. E.g. displacement: Grab that box and put it overthere.
2	Direct control level	Execute atomary commands like: Grab that box. Or: Put it over there.
1	Hardwired locking level	Maintain safety by means of hard wired negative (often tactile) sensors. No signal means: Stop .
0	Inherent safety level	Physical prevention of accidents: Arm not long enough to reach operator. Concrete buffer at the end of the track.

It's the levels 2 and 3 that the PLC programming style treated in the previous paragraphs is most suitable for. Level 4 is better tackled using mainstream object- or data-oriented programming. Level 1 is the domain of special purpose hard wired electrical circuits and a limited amount of simple and robust sensors. Level 0 is the level where the quay turns the ship.

Apart from the above hierarchy, there's a tendency for functionality to move from the control itself into the sensors and actuators. High speed feedback loops are part of the servo motors driving the robot's movements, rather than of the central program. The same holds e.g. for numerical integration of accelerometer data to obtain speed and position. As far as these intelligent actuators and sensors are robust, the trend towards decentralization is beneficial, since it replaces expensive and vulnerable special purpose software by out-of-the-box components. However it may be desirable for the central control to keep a watchful eye on these intelligent components, since the central control is the one able to combine all input information and draw conclusions, e.g. about sensor failure.

Keeping the possibility of sensor failure in mind, it may seem like a good idea to have lots of redundancy in the sensorical capabilities of the robot, but there's a downside to this. If sensors differ of opinion, what should the control do? Halt the system? But that often means halting production. The more sensors are present, the more sensor failure will occur. In some situations a minor chance on limited damage is to be preferred over a robot that regularly blocks a whole production line for no reason other than failure of a redundant sensor.